

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)
Toyohiko Yoshida, *et al.*) Group Art Unit: 2763
Application No.: 09/146,259) Examiner: unassigned
Filed: September 3, 1998)
For: DATA PROCESSING DEVICE)

#3
10/17/98

CLAIM FOR CONVENTION PRIORITY

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed:

Japanese Patent Application No. 10-027158

Filed: February 9, 1998.

In support of this claim, enclosed is a certified copy of said prior foreign application. Said prior foreign application was referred to in the oath or declaration. Acknowledgment of receipt of this certified copy is requested.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

Date: October 13, 1998

By: Ronni S. Jillions
Ronni S. Jillions
Registration No. 31,979

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1 9 9 8 年 2 月 9 日

出 願 番 号

Application Number:

平成 1 0 年特許願第 0 2 7 1 5 8 号

出 願 人

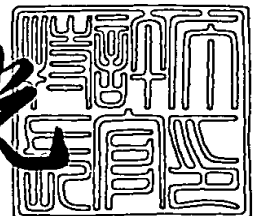
Applicant (s):

三菱電機株式会社

1 9 9 8 年 3 月 1 3 日

特 許 庁 長 官
Commissioner,
Patent Office

荒井寿光



出証番号 出証特平 1 0 - 3 0 1 4 7 1 4

【書類名】 特許願

【整理番号】 50932401

【提出日】 平成10年 2月 9日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/38

【発明の名称】 データ処理装置

【請求項の数】 20

【発明者】

 【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

 【氏名】 吉田 豊彦

【発明者】

 【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

 【氏名】 藤井 秀至

【特許出願人】

 【識別番号】 000006013

 【氏名又は名称】 三菱電機株式会社

【代理人】

 【識別番号】 100102439

 【弁理士】

 【氏名又は名称】 宮田 金雄

【選任した代理人】

 【識別番号】 100103894

 【弁理士】

 【氏名又は名称】 家入 健

【選任した代理人】

 【識別番号】 100092462

 【弁理士】

【氏名又は名称】 高瀬 彌平

【手数料の表示】

【予納台帳番号】 011394

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704079

【書類名】 明細書

【発明の名称】 データ処理装置

【特許請求の範囲】

【請求項 1】 プログラムシーケンスに記述されたコード化された複数の演算命令を順次デコードして、各演算命令に対応する制御信号を出力する命令デコーダ、および、

前記命令デコーダから出力される前記制御信号に従って、前記複数の演算命令によりそれぞれ指定された演算を実行する命令実行部を備え、

第 1 の期間において、前記命令デコーダが前記複数の演算命令のうちの第 1 の演算命令をデコードして第 1 の制御信号を出力し、前記第 1 の期間に続く第 2 の期間において、前記命令実行部が前記第 1 の制御信号に従って前記第 1 の演算命令で指定された演算を実行し、

第 3 の期間において、前記命令デコーダが、前記複数の演算命令のうちの所定の条件の基で演算が実行される第 2 の演算命令をデコードして第 2 の制御信号を出力し、前記第 3 の期間の終わりから前記第 2 の期間と同じ時間、あるいはそれより長い時間が経過した後に開始される第 4 の期間において、前記命令実行部が前記所定の条件を満たしているか否かを判断し、その判断結果に応じて前記第 2 の演算命令で指定された演算を実行する、データ処理装置。

【請求項 2】 その保持される値を可変に設定できる遅延量指定レジスタを備え、

前記命令実行部は、前記遅延量指定レジスタに遅延量として保持されている値に応じて第 2 の演算命令が所定の条件を満たしているかの判断を開始する、請求項 1 に記載のデータ処理装置。

【請求項 3】 コード化された第 2 の演算命令は、演算を指定する演算指定フィールドと、第 3 の期間の終わりから第 4 の期間の開始までの間隔を指定する遅延量指定フィールドとを有し、

遅延量指定レジスタには、前記遅延量指定フィールドに記述された内容に従って遅延量が設定される、請求項 2 に記載のデータ処理装置。

【請求項4】 複数の演算命令の各々に対応付けられたアドレスを順次カウントして保持するプログラムカウンタを備え、

前記遅延量指定レジスタは遅延量としてアドレス値を保持し、

前記命令実行部は、前記遅延量指定レジスタに保持されたアドレス値が前記プログラムカウンタの値と一致したことに応答して、第2の演算命令が所定の条件を満たしているかの判断を開始する、請求項2に記載のデータ処理装置。

【請求項5】 命令実行部は、第4の期間に含まれる第5の期間において、所定の条件を満たしているか否かを判断し、前記第4の期間に含まれ、且つ前記第5の期間の終了して第2の期間と同じ時間あるいはそれより長い時間が経過して開始する第6の期間に前記所定の条件を満たしたときに第2の演算命令で指定された演算を実行する、請求項1に記載のデータ処理装置。

【請求項6】 各々はその保持される値を可変に設定できる第1および第2の遅延量指定レジスタを備え、

前記命令実行部は、前記第1の遅延量指定レジスタに第1の遅延量として保持されている値に従って第2の演算命令が所定の条件を満たしているかの判断を開始し、

前記第2の遅延量指定レジスタに第2の遅延量として保持されている値に従って第2の演算命令所定の条件を満たしたときに第2の演算命令で指定された演算を開始して実行する、請求項5に記載のデータ処理装置。

【請求項7】 コード化された第2の演算命令は、演算を指定する演算指定フィールドと、第3の期間の終わりから第4の期間の開始までの時間を指定する第1の遅延量指定フィールドと、第5の期間の終わりから第6の期間の開始までの時間を指定する第2の遅延量指定フィールドとを有し、

第1の遅延量指定レジスタには、前記第1の遅延量指定フィールドに記述された内容に従って第1の遅延量が設定され、第2の遅延量指定レジスタには、前記第2の遅延量指定フィールドに記述された内容に従って第2の遅延量が設定される請求項6に記載のデータ処理装置。

【請求項8】 複数の演算命令の各々に対応付けられたアドレスを順次カウントして保持するプログラムカウンタを備え、

前記第 1 および第 2 の遅延量指定レジスタはそれぞれ第 1 および第 2 の遅延量としてアドレス値を保持し、

前記命令実行部は、前記第 1 の遅延量指定レジスタに保持されたアドレス値が前記プログラムカウンタの値と一致したことに応答して、所定の条件が満たされているかの判断を開始し、前記第 2 の遅延量指定レジスタに保持されたアドレス値が前記プログラムカウンタの値と一致したことに応答して、前記所定の条件を満たした場合に第 2 の演算命令で指定された演算を開始し実行する、請求項 6 に記載のデータ処理装置。

【請求項 9】 命令デコーダは、第 3 の期間より後に開始される第 7 の期間において、複数の演算命令のうちの第 3 の演算命令をデコードして第 3 の制御情報を出力し、命令実行部は、前記第 7 の期間より後に開始される第 8 の期間において、前記第 3 の制御信号に従って前記第 3 の演算命令で指定された演算を実行してその演算結果を所定の記憶場所に書き込み、

第 2 の演算命令は、前記第 3 の演算命令の演算結果が所定の値を有している場合に実行される演算命令を指定しており、

前記命令実行部は、第 4 の期間の開始が少なくとも前記第 8 の期間の後になるように、前記所定の記憶場所を参照して前記第 2 の演算命令により指定される演算を実行すべきか否か決定する、請求項 1 に記載のデータ処理装置。

【請求項 10】 所定の記憶場所はフラグあるいはレジスタであり、

第 3 の演算命令は、2 つのレジスタの値を比較して、その比較結果を前記所定の記憶場所に書き込む比較命令である、請求項 9 に記載のデータ処理装置。

【請求項 11】 第 2 の演算命令は、分岐命令、ジャンプ命令、あるいは加算命令である、請求項 1 に記載のデータ処理装置。

【請求項 12】 複数の演算命令の各々は、演算の内容を指定する演算指定フィールドと、その演算の実行条件を指定する条件指定フィールドと、その実行条件を判断するタイミングを遅延させる量を指定する遅延量指定フィールドとを有し、

第 1 の演算命令は無条件に実行される命令であり、前記第 1 の演算命令の条件指定フィールドに無条件であることを示す情報が記述され、

第2の演算命令の条件指定フィールドおよび遅延量指定フィールドに、それぞれ所定の条件を示す情報および第3の期間の終わりから第4の期間の開始までの間隔を示す情報が記述され、

前記命令デコーダは、前記第1の演算命令における演算指定フィールドに従って第1の制御信号を出力し、前記第1の演算命令における条件指定フィールドに基づき、前記命令実行部が第2の期間で前記第1の演算命令で指定された演算を実行するように前記命令実行部を制御し、

前記命令デコーダは、前記第2の演算命令における演算指定フィールドに従って第1の制御信号を出力し、前記第2の演算命令における遅延量指定フィールドに従って第4の期間で所定の条件を満たしているかを判断させるように前記命令実行部を制御し、前記第2の演算命令における条件指定フィールドに従って所定の条件を満たしたか否かを決定するように前記命令実行部を制御する、請求項1に記載のデータ処理装置。

【請求項13】 複数の演算命令の各々は、各演算の内容を指定する演算指定フィールドと、その演算の実行条件を指定する条件指定フィールドと、その実行条件を判断するタイミングを遅延させる量を指定する遅延量指定フィールドとを有し、

第1の演算命令は条件付き命令であり、

前記第1の演算命令においては、前記条件指定フィールドおよび遅延量指定フィールドに、それぞれ第2の演算命令の所定の条件とは別の条件を示す情報、および第1の期間に前記別の条件を判断することを示す情報が記述され、

第2の演算命令において、前記条件指定フィールドおよび遅延量指定フィールドに、それぞれ所定の条件を示す情報および第3の期間の終わりから第4の期間の開始までの間隔を示す情報が記述されており、

前記命令デコーダは、前記第1の演算命令における演算指定フィールドに従って第1の制御信号を出力し、前記第1の演算命令における条件指定フィールドおよび遅延量指定フィールドに基づき、前記命令実行部が第2の期間で前記第1の演算命令で指定された演算を実行するように前記命令実行部を制御し、

前記命令デコーダは、前記第2の演算命令における条件指定フィールドおよび

演算指定フィールドに従って第1の制御信号を出力し、前記第2の演算命令における遅延量指定フィールドに従って第4の期間で所定の条件を満たしているかを判断させるように前記命令実行部を制御し、前記第2の演算命令における条件指定フィールドに従って所定の条件を満たしたか否かを決定するように前記命令実行部を制御する、請求項1に記載のデータ処理装置。

【請求項14】 プログラムシーケンスに記述されたコード化された複数の演算命令を順次デコードして、各演算命令に対応する制御信号を出力する命令デコーダ、および、

前記命令デコーダから出力される前記制御信号に従って、前記複数の演算命令によりそれぞれ指定された演算を実行する命令実行部を備え、

前記複数の演算命令のうちの1つが条件付きの演算命令である場合、

前記命令デコーダは、前記条件付きの演算命令をデコードして第1の制御信号を出力し、

前記命令実行部は、

前記命令デコーダから出力される第1の制御信号を保持させる第1のレジスタ、前記条件付き演算命令に指定される演算が実行される条件を示す第1の情報を保持させる第2のレジスタ、および、前記条件の判断を開始する時期を示す第2の情報を保持する第3のレジスタを含み、

前記命令実行部は、

前記第3のレジスタに保持された第2の情報に基づき、前記条件の判断が開始される時期であることを検出したことに応答して、前記第2のレジスタに保持された第1の情報に基づき前記条件が満たされているか否かの判断を開始し、その判断結果に応じて前記第1のレジスタに保持された第1の制御信号を読み出し、その読み出された第1の制御信号に従い前記演算命令に指定された演算の実行を開始するデータ処理装置。

【請求項15】 第3のレジスタに保持される第2の情報を可変に設定することのできる、請求項14に記載のデータ処理装置。

【請求項16】 複数の演算命令の各々に対応付けられたアドレスを順次カウントして保持するプログラムカウンタを備え、

前記第3のレジスタには、第2の情報としてアドレス値が保持され、

前記命令実行部は、前記第3のレジスタに保持されたアドレス値が前記プログラムカウンタのアドレスと一致したことを検出し、その検出に応答して前記条件が満たされているか否かの判断を開始する、請求項14に記載のデータ処理装置

。【請求項17】 コード化された条件付き演算命令は、演算の内容を指定する演算指定フィールドと、その演算の実行条件を指定する条件指定フィールドと、その実行条件を判断する時期を指定する遅延量指定フィールドとを有し、

前記命令デコーダは、前記演算指定フィールドに記述された内容に基づいて前記第1の制御信号を生成し、前記条件指定フィールドに記述された内容を第1の情報として出力するとともに前記遅延量指定フィールドに記述された内容を出力し、

前記命令デコーダから出力された第1の情報は、前記第2のレジスタに保持され、

前記命令実行部は、前記命令デコーダから出力される前記遅延量指定フィールドに従って第2の情報を第3のレジスタに書き込む、請求項15に記載のデータ処理装置。

【請求項18】 命令実行部は、

さらに、演算命令で指定された演算を開始する時期を示す第3の情報を保持する第4のレジスタを有し、

この第3の情報に従って、前記演算命令で指定された演算が開始される時期であることを検出し、その検出結果に応答して条件が満たされているか否かを判断し、その判断結果に応じて前記演算命令で指定された演算を開始する、請求項14に記載のデータ処理装置。

【請求項19】 複数の演算命令の各々に対応付けられたアドレスを順次カウントして保持するプログラムカウンタを備え、

前記第3のレジスタには、第2の情報としてアドレス値が保持され、

前記第4のレジスタには、第3の情報として前記第2の情報とは異なるアドレス値が保持され、

前記命令実行部は、前記第3のレジスタに保持されたアドレス値が前記プログラムカウンタのアドレスと一致したことを検出し、その検出に応答して前記条件が満たされているか否かの判断を開始し、前記第4のレジスタに保持されたアドレス値が前記プログラムカウンタのアドレスと一致したことを検出し、その検出に応答して前記演算命令に指定された演算の実行を開始する、請求項18に記載のデータ処理装置。

【請求項20】 コード化された条件付き演算命令は、演算の内容を指定する演算指定フィールドと、その演算の実行条件を指定する条件指定フィールドと、その実行条件を判断する時期を指定する第1の遅延量指定フィールドと、その演算実行を開始する時期を指定する第2の遅延量指定フィールドを有し、

前記命令デコーダは、前記演算指定フィールドに記述された内容に基づいて前記第1の制御信号を生成し、前記条件指定フィールドに記述された内容を第1の情報として出力するとともに前記第1および第2の遅延量指定フィールドに記述された内容を出し、

前記命令実行部は、前記命令デコーダから出力された前記第1の遅延量指定フィールドに記述された内容に従って第2の情報を第3のレジスタに書き込み、さらに前記命令デコーダから出力された前記第2の遅延量指定フィールドに記述された内容に従って第3の情報を第4のレジスタに書き込む、請求項18に記載のデータ処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、プログラムシーケンスにおいて条件付き演算命令をうまくスケジューリングするためのデータ処理装置に関するものである。

【0002】

【従来の技術】

従来、パイプラインにより複数の命令を順々に実行するマイクロプロセッサにおいて、分岐命令の実行はパイプラインを乱してプロセッサの処理効率を低下させる1つの原因となっている。この処理性能の低下を防止するために、例えば遅

延スロットを用いた手法がある。

【0003】

プログラマーが次の命令シーケンス 1 を作成したとする。

〈命令シーケンス 1〉

```
(I 0)  CMPEQ r10,r11
(I 1)  ADD r1,r2
(I 2)  BRA F0=1 H■1000
(I 3)  . . . . .
(I 4)  . . . . .
```

【0004】

命令 I 0 は、レジスタ 10 とレジスタ 11 とを比較して等しければフラグ F0 をセットする比較命令である。命令 I 1 は、レジスタ r1 とレジスタ r2 との内容を加算してレジスタ r1 に書き込む加算命令である。命令 I 2 は、フラグ F0 がセットされていたら、メモリの 1000 番地にある命令に分岐する条件分岐命令である。命令 I 3、I 4 は任意の命令で、命令 I 2 が実行される時点ではすでにマイクロプロセッサに投入されている。命令 I 2 の実行により分岐が発生したらパイプライン処理中の命令 I 3、I 4 は無効化される。

【0005】

そこで、命令 I 1 は、命令 I 2 の分岐条件を決定する演算を行う命令ではなく、命令 I 2 の分岐に拘わらず実行される命令であることを考慮して、次の命令シーケンス 2 のような命令のスケジューリングを行う。

〈命令シーケンス 2〉

```
(I 0)  CMPEQ r10,r11
(I 2)  BRA F0=1 H■1000
(I 1)  ADD r1,r2
(I 5)  NOP
```

【0006】

命令 I 2 の実行により 1000 番地への分岐が決定されたとしても、すでにパイプラインに投入され、処理されている命令 I 1 は無効にすることなく実行すればよ

く、命令I2の実行される時点で2つ命令がパイプラインに投入されているアーキテクチャの場合、遅延スロットは2つの命令で埋められることになる。命令シーケンス2の場合、遅延スロットは命令I1、I5により構成される。命令I5はいわゆるNOP (no operation) 命令である。命令I2により分岐が決定されると命令I5の次には分岐先の命令がフェッチされる。

この命令シーケンス1から命令シーケンス2へのスケジューリングはプログラム自身あるいはコンパイラにより行われる。

このような命令のスケジューリングに関して、“Computer Architecture: A Quantitative Approach, Morgan Kaufmann 社, 1990年”で種々の手法が提案されている。

【0007】

【発明が解決しようとする課題】

このように、マイクロプロセッサの処理能力を最大に発揮させるために、プログラム上で命令をスケジューリングする技術は重要である。しかしながら命令の種類によってスケジューリングする上で種々の制約が存在する。上記の例によると、命令シーケンスにおいて、命令I2の条件分岐命令はその実行条件を決定する命令I0より前に置くことはできない。これは、命令I2が分岐の条件であるフラグF0の内容を参照する際、まだその条件を決定する命令I0の実行が終了できないからである。このようにして、条件付きの分岐命令に限らず条件付きの算術演算命令も同様に命令のスケジューリングの自由度を損ねる原因となる。

【0008】

VLIW (very long instruction word) タイプのマイクロプロセッサにおいては、並列に実行することのできる複数のサブ命令を1つの命令セットで表現している。このタイプではどの命令が並列に実行できるかを考慮した高度なスケジューリング技術が必要となり、条件付きの演算命令の存在により、いわゆるNOP (No Operation) 命令というプログラム上意味のない命令を命令シーケンスに挿入しなければならないことが多くなる。NOP (No Operation) 命令を処理することもマイクロプロセッサの処理性能を低下させる原因でもある。

【0009】

しかるに、本発明の技術課題は、条件付きの演算命令を処理するデータ処理装置において、自由度の高い命令のスケジューリングができる環境をプログラマに提供することである。

【0010】

【課題を解決するための手段】

この発明によるデータ処理装置は、プログラムシーケンスに記述されたコード化された複数の演算命令を順次デコードして、各演算命令に対応する制御信号を出力する命令デコーダ、および、この命令デコーダから出力される制御信号に従って、複数の演算命令によりそれぞれ指定された演算を実行する命令実行部を備えている。第1の演算命令に関して、第1の期間でデコードされ、これに続く第2の期間でその指定される演算が実行される。一方、所定の条件の基で演算が実行される第2の演算命令に関して、第3の期間でデコードされ、第3の期間の終わりから第2の期間と同じ時間、あるいはそれより長い時間が経過した後に開始される第4の期間において、命令実行部が所定の条件を満たしているか否かを判断し、その判断結果に応じて第2の演算命令で指定された演算を実行している。

【0011】

データ処理装置は、さらに、保持される値を可変に設定できる遅延量指定レジスタを備えている。命令実行部は前記遅延量指定レジスタに遅延量として保持されている値に応じて第2の演算命令が所定の条件を満たしているかの判断を開始している。

【0012】

コード化された第2の演算命令は、演算を指定する演算指定フィールドと、第3の期間の終わりから第4の期間の開始までの間隔を指定する遅延量指定フィールドとを有し、遅延量指定レジスタには、遅延量指定フィールドに記述された内容に従って遅延量が設定される。

【0013】

データ処理装置は、さらに、複数の演算命令の各々に対応付けられたアドレスを順次カウントして保持するプログラムカウンタを備えている。遅延量指定レ

スタは遅延量としてアドレス値を保持するものであり、命令実行部は、遅延量指定レジスタに保持されたアドレス値がプログラムカウンタの値と一致したことに応答して、第2の演算命令が所定の条件を満たしているかの判断を開始している。

【0014】

命令実行部は第4の期間に含まれる第5の期間において、所定の条件を満たしているか否かを判断する。そして命令実行部は第4の期間に含まれており且つこの第5の期間の終了して第2の期間と同じ時間あるいはそれより長い時間が経過して開始する第6の期間に所定の条件を満たしたときに第2の演算命令で指定された演算を実行している。

【0015】

データ処理装置は、各々はその保持される値を可変に設定できる第1および第2の遅延量指定レジスタを備えている。命令実行部は、第1の遅延量指定レジスタに第1の遅延量として保持されている値に従って第2の演算命令が所定の条件を満たしているかの判断を開始し、第2の遅延量指定レジスタに第2の遅延量として保持されている値に従って第2の演算命令所定の条件を満たしたときに第2の演算命令で指定された演算を開始して実行している。

【0016】

コード化された第2の演算命令は、演算を指定する演算指定フィールドと、第3の期間の終わりから第4の期間の開始までの時間を指定する第1の遅延量指定フィールドと、第5の期間の終わりから第6の期間の開始までの時間を指定する第2の遅延量指定フィールドとを有している。第1の遅延量指定レジスタには、第1の遅延量指定フィールドに記述された内容に従って第1の遅延量が設定され、第2の遅延量指定レジスタには、第2の遅延量指定フィールドに記述された内容に従って第2の遅延量が設定される。

【0017】

第1および第2の遅延量指定レジスタはそれぞれ第1および第2の遅延量としてアドレス値を保持している。命令実行部は第1の遅延量指定レジスタに保持されたアドレス値がプログラムカウンタの値と一致したことに応答して、所定の条

件が満たされているかの判断を開始している。さらに命令実行部は、第2の遅延量指定レジスタに保持されたアドレス値がプログラムカウンタの値と一致したことに応答して、所定の条件を満たした場合に第2の演算命令で指定された演算を開始し実行している。

【0018】

複数の演算命令のうち第3の演算命令に関し、第3の期間より後に開始される第7の期間において命令デコーダでデコードされ、前記第7の期間より後に開始される第8の期間において第3の演算命令で指定された命令実行部で演算が実行されその演算結果が所定の記憶場所へ書き込まれる。このとき、第2の演算命令は、第3の演算命令の演算結果が所定の値を有している場合に実行される演算命令を指定しており、命令実行部は、第4の期間の開始が少なくとも第8の期間の後になるように、所定の記憶場所を参照して演算を実行すべきか否か決定している。

【0019】

所定の記憶場所はフラグあるいはレジスタである。また第3の演算命令は、2つのレジスタの値を比較して、その比較結果を前記所定の記憶場所へ書き込む比較命令である。さらに、第2の演算命令は、分岐命令、ジャンプ命令、あるいは加算命令である。

【0020】

複数の演算命令の各々は、演算の内容を指定する演算指定フィールドと、その演算の実行条件を指定する条件指定フィールドと、その実行条件を判断するタイミングを遅延させる量を指定する遅延量指定フィールドとを有している。

【0021】

第1の演算命令が無条件に実行される命令であれば、第1の演算命令の条件指定フィールドは無条件であることを示す情報が記述されている。命令デコーダは、第1の演算命令における演算指定フィールドに従って第1の制御信号を出力し、条件指定フィールドに基づき、第2の期間で前記第1の演算命令で指定された演算を実行するように前記命令実行部を制御する。

【0022】

また、第1の演算命令が条件付き演算命令であれば条件指定フィールドおよび遅延量指定フィールドに、それぞれ第2の演算命令の所定の条件とは別の条件を示す情報、および第1の期間に前記別の条件を判断することを示す情報が記述されている。命令デコーダは、第1の演算命令における演算指定フィールドに従って第1の制御信号を出力し、条件指定フィールドおよび遅延量指定フィールドに基づき、第2の期間で第1の演算命令で指定された演算を実行するように命令実行部を制御する。

命令デコーダは、第1の演算命令における条件指定フィールドおよび遅延量指定フィールドに従って条件を満たしているか否かを判断し、その判断結果に応じて第1の演算命令における演算指定フィールドに従って第1の制御信号を出力する。

【0023】

第2の演算命令の条件指定フィールドおよび遅延量指定フィールドに、それぞれ所定の条件を示す情報および第3の期間の終わりから第4の期間の開始までの間隔を示す情報が記述されている。命令デコーダは、第2の演算命令における演算指定フィールドに従って第1の制御信号を出力し、第2の演算命令における遅延量指定フィールドに従って第4の期間で所定の条件を満たしているかを判断させるように命令実行部を制御し、第2の演算命令における条件指定フィールドに従って所定の条件を満たしたか否かを決定するように命令実行部を制御する。

【0024】

また、この発明によるデータ処理装置は、条件付きの演算命令をデコードして第1の制御信号を出力する命令デコーダ、その第1の制御信号に従い演算を実行する命令実行部を備えている。命令実行部は第1の制御信号を保持させる第1のレジスタ、条件付き演算命令に指定される演算が実行される条件を示す第1の情報を保持させる第2のレジスタ、および、その条件の判断を開始する時期を示す第2の情報を保持する第3のレジスタを含んでいる。そして命令実行部は、第3のレジスタに保持された第2の情報に基づき、条件の判断が開始される時期であることを検出したことに応答して第2のレジスタに保持された第1の情報に基づ

き前記条件が満たされているか否かの判断を開始し、その判断結果に応じて第1のレジスタに保持された第1の制御信号を読み出して演算命令に指定された演算の実行を開始する。

【0025】

第3のレジスタに保持される第2の情報を可変に設定することができる。

【0026】

データ処理装置は、複数の演算命令の各々に対応付けられたアドレスを順次カウントして保持するプログラムカウンタを備えている。第3のレジスタには、第2の情報としてアドレス値が保持され、命令実行部は第3のレジスタに保持されたアドレス値がプログラムカウンタのアドレスと一致したことを検出し、その検出に応答して条件が満たされているか否かの判断を開始している。

【0027】

条件付き演算命令は、演算の内容を指定する演算指定フィールドと、その演算の実行条件を指定する条件指定フィールドと、その実行条件を判断する時期を指定する遅延量指定フィールドとを有する。命令デコーダは演算指定フィールドに記述された内容に基づいて第1の制御信号を生成し、条件指定フィールドに記述された内容を第1の情報として出力するとともに遅延量指定フィールドに記述された内容を出力する。命令デコーダから出力された第1の情報は第2のレジスタに保持される。そして命令実行部は、命令デコーダから出力される遅延量指定フィールドに従って第2の情報を第3のレジスタに書き込む。

【0028】

命令実行部は、さらに、演算命令で指定された演算を開始する時期を示す第3の情報を保持する第4のレジスタを有している。命令実行部はこの第3の情報に従って、演算命令で指定された演算が開始される時期であることを検出し、その検出結果に応答して条件が満たされているか否かを判断し、さらにその判断結果に応じて前記演算命令で指定された演算を開始する。

【0029】

データ処理装置は、複数の演算命令の各々に対応付けられたアドレスを順次カウントして保持するプログラムカウンタを備えている。第3のレジスタには、第

2の情報としてアドレス値が保持され、第4のレジスタには、第3の情報として第2の情報とは異なるアドレス値が保持されている。命令実行部は、第3のレジスタに保持されたアドレス値がプログラムカウンタのアドレスと一致したことを検出し、その検出に応答して条件が満たされているか否かの判断を開始する、さらに命令実行部は第4のレジスタに保持されたアドレス値がプログラムカウンタのアドレスと一致したことを検出し、その検出に応答して前記演算命令に指定された演算の実行を開始している。

【0030】

コード化された条件付き演算命令は、演算の内容を指定する演算指定フィールドと、その演算の実行条件を指定する条件指定フィールドと、その実行条件を判断する時期を指定する第1の遅延量指定フィールドと、その演算実行を開始する時期を指定する第2の遅延量指定フィールドを有している。命令デコーダは、演算指定フィールドに記述された内容に基づいて第1の制御信号を生成し、条件指定フィールドに記述された内容を第1の情報として出力するとともに第1および第2の遅延量指定フィールドに記述された内容を出力する。命令実行部は、命令デコーダから出力された第1の遅延量指定フィールドに記述された内容に従って第2の情報を第3のレジスタに書き込み、さらに命令デコーダから出力された第2の遅延量指定フィールドに記述された内容に従って第3の情報を第4のレジスタに書き込む。

【0031】

【発明の実施の形態】

以下、この発明の実施の一形態を説明する。

実施の形態1.

図1はこの発明の実施の形態1によるマイクロプロセッサの構成を示すブロック図である。このマイクロプロセッサは32ビットの内部データバスを有する32ビットマイクロプロセッサである。図において、2は命令RAM 6から64ビット幅のIDバスを介して入力した命令コードをデコードする処理を行う命令デコードユニット（命令デコーダ）、3はアドレス計算を行うメモリユニット（命令実行部）、4は論理演算やシフト演算を行う整数演算ユニット（命令実行部）

、5は32ビット×64ワードの汎用レジスタ、7はデータが格納されるデータRAMである。

【0032】

命令デコードユニット2において、8、9はそれぞれ命令コードをデコードするデコーダ、10はプロセッサの状態を示すプロセッサ状態語 (Processor Status Word、以下、プロセッサ状態語をPSWと呼ぶ) である。命令デコードユニット2は、さらに、デコーダ8のデコード結果とPSW10の内容にもとづいて制御信号11を作成し、それをメモリユニット3に与える。また、命令デコードユニット2は、デコーダ9のデコード結果とPSW10の内容にもとづいて制御信号12を作成し、それを整数演算ユニット4に与える。

【0033】

メモリユニット3において、13はジャンプや分岐を含まない命令を実行するとPC値に8を加えて次に実行する命令に対するPC値を算出するとともに、ジャンプや分岐を含む命令の実行時に分岐変位をPC値に加算したり、演算で指定されたアドレッシングモードに応じた計算を行ってジャンプ先の命令に対するPC値を計算するPC制御部である。また、PC制御部13は、計算したPC値を32ビット幅のIAバスを介して命令RAM6に送り、命令RAM6から命令コードを出力させる。14はオペランドとなるデータのアクセスを制御するメモリ制御部である。メモリ制御部14は、32ビット幅のDAバスを介してアドレスデータをデータRAM7に転送し命令実行に必要なデータを64ビット幅のDDバスを介してアクセスする。15は汎用レジスタ5から32ビット幅のS1バス、S2バス、S3バスを介して転送された最大3ワードのデータを用いて算術論理演算を行い演算結果を32ビット幅のD1バスを介して汎用レジスタ5に転送するALU、16は汎用レジスタ5からS1バス、S2バス、S3バスを介して転送されたデータを用いてシフト演算を行い演算結果をD1バスを介して汎用レジスタ5に転送するシフタである。

【0034】

メモリユニット3に対して、S1バス、S2バス、S3バス、S4バスを介して、32ビット長のデータを一時に4ワード転送することが可能である。従って

、例えば、第1のレジスタの内容と第2のレジスタの内容との和でアドレッシングされるメモリの領域に第3のレジスタの内容をストアするとともに、第3のレジスタの内容をストアしたアドレスに所定値を加算して得られる値でアドレッシングされるメモリの領域に第4のレジスタの内容をストアする2ワードストア命令を実現することができる。また、メモリユニット3は、D1バスおよびD2バスを介して、メモリユニット3内での2ワードの演算結果またはデータRAM7から転送された2ワードのデータを汎用レジスタ5に転送することができる。

【0035】

また、PC制御部13はレジスタ30、31、32および33、さらに1ビットを保持する記憶回路34を含む。メモリ制御部14はレジスタ40、41および42を含む。ALU15はレジスタ50、51および52を含む。さらにシフタ16はレジスタ60、61および62を含む。これらのレジスタに関しては後述する。

【0036】

整数演算ユニット4において、17は汎用レジスタ5から32ビット幅のS4バス、S5バス、S6バスを介して転送された最大3ワードのデータを用いて乗算を行い演算結果を32ビット幅のD2バス、D3バスを介して汎用レジスタ5に転送する乗算器、18は乗算の結果を累積加算または累積減算して保持するアキュムレータである。アキュムレータとして、64ビットのものが2本ある。19は汎用レジスタ5からS4バス、S5バス、S6バスを介して転送された最大3ワードのデータを用いて算術論理演算を行い演算結果をD2バス、D3バスを介して汎用レジスタ5に転送するALU、20は汎用レジスタ5からS4バス、S5バス、S6バスを介して転送されたデータを用いてシフト演算を行い演算結果をD2バス、D3バスを介して汎用レジスタ5に転送するシフタである。

【0037】

乗算器17はレジスタ70、71、および72を含む。ALU19はレジスタ80、81および82を含む。さらにシフタ20はレジスタ90、91および92を含む。これらのレジスタに関しても後述する。

【0038】

このマイクロプロセッサでは、汎用レジスタ5から、最大6種類のレジスタ値を読み出すことが可能であって、読み出されたデータは、それぞれ、S1バス、S2バス、S3バス、S4バス、S5バス、S6バスに出力される。また、汎用レジスタ5には、D1バス、D2バス、D3バスを介して最大3種類のレジスタ値を同時に書き込むことが可能である。

【0039】

図2はマイクロプロセッサ1の命令フォーマットを示す説明図である。命令フォーマットとして、図2(a)に示すような1つの命令コードで2つの演算(operation)を指示する2演算命令のフォーマット101と、図2(b)に示すような1つの命令コードで1つの演算を指示する1演算命令のフォーマット102とがある。

2演算命令のフォーマット101には、フィールド103およびフィールド104からなるフォーマットフィールドと、2つの演算フィールド106、107と、演算フィールド106、107にそれぞれ付属する実行条件フィールド401および402と、さらに、実行条件フィールド401および402にそれぞれ付属する条件判定遅延量指定フィールド(以下、CDフィールド)404、405とが含まれる。

1演算命令のフォーマット102には、フィールド103および104からなるフォーマットフィールドと、フィールド108、109および110からなる演算フィールドと、この演算フィールドに付属する実行条件フィールド403と、この実行条件フィールド403に付属するCDフィールド406とが含まれる。

【0040】

フォーマットフィールドは、以下のような意味を示す。

コード：フォーマット	実行の順番	
	operation_0	operation_1
FM=00：2命令	第1	第1
01：2命令	第1	第2
10：2命令	第2	第1

11:1命令

第1

—

ここで、FMは、フィールド103およびフィールド104からなる2ビットの値である。

【0041】

マイクロプロセッサ1の複数のパイプラインステージは、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M、ライトバックステージWにより構成され、各々のステージでの処理は1クロックサイクルでなされる。

図3はマイクロプロセッサ1において2演算命令101を処理するパイプラインステージを説明するための説明図である。命令フェッチユニットIFにおいて、2演算命令101がメモリRAM6から命令デコードユニット2へフェッチされる。命令デコードユニットD/Aにおいて、演算フィールド106に記述されたoperation_0がデコーダ8でデコードされ、演算フィールド107に記述されたoperation_1がデコーダ9でデコードされる。また、operation_0、1の各オペランドアドレス、又は、operation_0、1の各々が分岐命令であれば分岐先アドレスが、命令デコードステージD/Aで算出される。命令実行ステージE/Mにおいて、制御信号11に従ってoperation_0で指定された演算がメモリユニット3で実行され、制御信号12に従ってoperation_1で指定された演算が整数演算ユニット4で実行される。operation_1がロード命令、ストア命令等のメモリアクセスを伴う命令であれば、命令実行ステージE/Mにおいてメモリユニット3によりメモリへのアクセスが行われる。ライトバックステージWにおいて、メモリユニット3で得られた演算結果、整数演算ユニット4で得られた演算結果が、それぞれoperation_0、1で指定されるレジスタに書き込まれる。演算結果をプロセッサ1内のレジスタに書き込む命令を伴わない命令（分岐命令、ジャンプ命令、メモリヘデータをストアするストア命令、演算結果をフラグに反映させる比較命令等）は、ライトバックステージWは存在しない。なおマイクロプロセッサによっては、ライトバックステージWが命令実行ステージE/Mと同じクロックサイクルで処理されるものもある。

【0042】

FM=00の場合、図3(a)に示すようにoperation_0、1のステージIF、D/

A、E/M、Wは互いに並列に行われ、operation_0、1は4クロックで処理される。

FM=01の場合、図3(b)に示すようにoperation_0のステージIF、D/A、E/M、Wは4クロックを要して連続的に行われる。operation_0、1のステージIF、D/Aは互いに並列に行われる一方、operation_1のステージE/M、Wは、operation_0より1クロック遅れて行われる。operation_1のステージE/Mがoperation_0のステージWと並列に行われる。よってoperation_1は5クロックで処理される。

FM=01の場合、図3(c)に示すようにoperation_1のステージIF、D/A、E/M、Wは4クロックを要して連続的に行われる。operation_0、1のステージIF、D/Aは互いに並列に行われる一方、operation_0のステージE/M、Wは、operation_1より1クロック遅れて行われる。operation_0のステージE/Mがoperation_1のステージWと並列に行われる。よってoperation_1は5クロックで処理される。

【0043】

また、図2(b)に示すフォーマットによる1演算命令102もまた、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M、ライトバックステージWにおいてそれぞれ1クロックサイクルでもって処理される。ステージIFにおいて、1演算命令102が命令RAM6から命令デコードユニット2へフェッチされる。ステージD/Aにおいて、1演算命令102がデコーダ8、9の各々に入力される。1演算命令102で指定される演算の種類に応じて、デコーダ8、9のいずれか一方が1演算命令102をデコードする。デコーダ8がデコードしたならば制御信号11を出力し、デコーダ9がデコードしたならば制御信号12を出力する。ステージE/Mにおいて、制御信号11（または12）に従ってメモリユニット3（又は整数演算ユニット4）が、1演算命令102で指定された演算を実行する。ステージWにおいて、ステージE/Mで得られた演算結果が、1演算命令102で指定されたレジスタに書き込まれる。

【0044】

次いで、実行条件コードについて説明する。実行条件フィールド401、402、403の各々は、以下のような意味を持つ。

コード： 実行条件

CC=000： 常時

001: F0=真 かつ F1=無視
 010: F0=偽 かつ F1=無視
 011: F0=無視 かつ F1=真
 100: F0=無視 かつ F1=偽
 101: F0=真 かつ F1=真
 110: F0=真 かつ F1=偽
 111: 予約済

【0045】

実行条件フィールド401～403の各々は、その付属する演算命令の実行条件を指定するもので、マイクロプロセッサの実行コントロールフラグF0、F1の値に応じて、演算フィールド106、107のoperation_0の演算やoperation_1の演算、およびフィールド108、109、110からなる演算フィールドの演算が有効であるか無効であるか定める。実行コントロールフラグF0、F1については後で説明するようにプロセッサステータスワード（PSW）10内にある。演算が有効であるとは、演算結果がレジスタ、メモリおよびフラグに反映され、その演算による動作の結果が残ることを意味する。一方演算が無効であるとは、演算命令がデコードされたがその指定された演算が実行されないこと、または演算されてもその演算結果がレジスタ、メモリおよびフラグに反映されず、あたかも無効演算（NOP）が実行されたかのような動作の結果が残ることを意味する。

【0046】

実行条件フィールドの値CC=000のときには、実行コントロールフラグF0、F1の値にかかわらず常に演算は有効である。CC=001のときには、実行コントロールフラグF0=真のときにのみ演算は有効である。実行コントロールフラグF1の状態はいつでもよい。CC=010のときには、実行コントロールフラグF0=偽のときにのみ演算は有効である。実行コントロールフラグF1の状態はいつでもよい。CC=011のときには、実行コントロールフラグF1=真のときにのみ演算は有効である。実行コントロールフラグF0の状態はいつでもよい。CC=100のときには、実行コントロールフラグF1=偽の

ときにのみ演算は有効である。実行コントロールフラグF0の状態はいずれでもよい。CC=101のときには、実行コントロールフラグF0=真かつF1=真のときにのみ演算は有効である。CC=110のときには、実行コントロールフラグF0=真かつF1=偽のときにのみ演算は有効である。CC=111のときの動作は未定義であり、ユーザはCC=111となる命令を用いることはできない。

【0047】

図4は演算フィールドの詳細な内容を示す説明図である。

フォーマット111～117は、それぞれ28ビットで表現される短型の演算フィールド106または演算フィールド107によるものである。フォーマット118は、フィールド108、109、110で構成される長型の演算フィールドによるものである。

【0048】

フォーマット111 (Short_M) は、演算内容を指定するフィールド120、レジスタ番号を指定する2つのフィールド121、122、レジスタ番号または6ビット長の即値を指定するフィールド123、およびフィールド123がレジスタ番号を示すのか即値を示すのかを指定するフィールド124で構成される。図3に示すように、フィールド124の値Xが「00」、「01」または「11」であるときにはフィールド123がレジスタ番号を示していることを示し、「10」であるときには即値を示していることを示す。

このフォーマット111は、レジスタ間接アドレッシングのメモリアクセス演算に用いられる。

【0049】

フォーマット112 (Short_A) は、演算内容を指定するフィールド120、レジスタ番号を指定する2つのフィールド121、122、レジスタ番号または6ビット長の即値を指定するフィールド123、およびフィールド123がレジスタ番号を示すのか即値を示すのかを指定するフィールド125で構成される。図3に示すように、フィールド125の値X'が「0」であるときにはフィールド123がレジスタ番号を示していることを示し、「1」であるときには

即値を示していることを示す。

このフォーマット 112 は、算術演算、論理演算、シフト演算およびビット演算に用いられる。

【0050】

フォーマット 113 (Short__B1) は、演算内容を指定するフィールド 120 およびレジスタ番号を指定するフィールド 126 で構成される。このフォーマット 113 は、レジスタ指定によるジャンプ命令および分岐命令に用いられる。

フォーマット 114 (Short__B2) は、演算内容を指定するフィールド 120 および 18 ビット長のディスプレイメントのフィールド 127 で構成される。このフォーマット 114 は、ジャンプ命令および分岐命令に用いられる。

【0051】

フォーマット 115 (Short__B3) は、演算内容を指定するフィールド 120、レジスタ番号を指定するフィールド 121、レジスタ番号または 12 ビット長の即値を指定するフィールド 128、フィールド 128 がレジスタ番号を示すのか即値を示すのかを指定するフィールド 129、およびゼロ判定にもとづいてフィールド 121 にもとづく条件ジャンプまたは条件分岐を行うか否か指定するフィールド 130 で構成される。

このフォーマット 115 は、条件ジャンプ命令および条件分岐命令に使用される。

【0052】

フォーマット 116 (Short__D1) は、演算内容を指定するフィールド 120、レジスタ番号を指定するフィールド 121、レジスタ番号または 12 ビット長の即値を指定するフィールド 128、フィールド 128 がレジスタ番号を示すのか即値を示すのかを指定するフィールド 129 で構成される。

このフォーマット 116 は、条件ジャンプ命令、条件分岐命令およびリピート命令に使用される。

フォーマット 117 (Short__D2) は、演算内容を指定するフィールド 120、レジスタ番号または 12 ビット長の即値を指定するフィールド 128、

フィールド 128 がレジスタ番号を示すのか即値を示すのかを指定するフィールド 129、遅延命令（ディレイド命令）に関するフィールド 131 で構成される。

このフォーマット 117 は、ディレイドジャンプ命令、ディレイド分岐命令およびリピート命令に使用される。

【0053】

フォーマット 118（Long）は、演算内容を指定するフィールド 120、レジスタ番号を指定する 2 つのフィールド 121、122、32 ビット長の即値を指定するフィールド 132 で構成される。

このフォーマット 118 は、複雑な算術演算、大きな即値を用いる算術演算、大きなディスプレイスメント付きレジスタ間接アドレッシングのメモリアクセス演算、大きな変位の分岐演算および絶対番地へのジャンプ命令などに使用される。

【0054】

図 5 はマイクロプロセッサのレジスタ構成を示す説明図である。

このマイクロプロセッサは、図 5（a）に示すような 64 本の 32 ビット長の汎用レジスタ 5、図 5（b）に示すような 12 本の制御レジスタ 150、および図 5（c）に示すような 2 本のアキュムレータ 18 を持つ。R0 の汎用レジスタ 140 の内容は常に 0 であり、そこへの書き込みは無視される。R62 の汎用レジスタは、サブルーチンからの戻り先アドレスが設定されるリンクレジスタである。R63 の汎用レジスタは、スタックポインタであり、PSW10 の SM フィールドの値に応じてユーザスタックポインタ（SPU）または割り込みスタックポインタ（SPI）として動作する。制御レジスタ 150 には、プログラムカウンタ 151、PSW10、および各種の専用レジスタが含まれる。

図 4 に示すフォーマット 112 による演算では、64 本の汎用レジスタ 5 のそれぞれを上位 16 ビットと下位 16 ビットとに分けてアクセスできる。また、2 本のアキュムレータ 18 を、上位 32 ビットと下位 32 ビットとに分けて別々にアクセスできる。

【0055】

図6はPSW10の詳細内容を示す説明図である。

PSW10の上位16ビットには、スタックポインタを切り替えるためのSMフィールド171、セルフデバグトラップ(SDBT)の検出を示すEAフィールド172、SDBTの許可を指定するDBフィールド173、割り込み許可を指定するIEフィールド174、リピート動作の許可を指定するRPフィールド175、モジュロアドレッシングの許可を指定するMDフィールド176がある。下位16ビットはフラグフィールド180である。フラグフィールド180には8個のフラグがあり、その中のF0フラグ181およびF1フラグ182は演算の有効/無効を指定する。各フラグの値は比較演算や算術演算の結果に応じて変化する。また、フラグ初期化演算で初期化したり、フラグ値書き込み演算で任意の値をフラグフィールド180に書き込むことによって変化する。フラグフィールド180の内容は、フラグ値読み出し演算によって読み出される。

【0056】

各フラグは、以下のような意味を有する。

SM=0	: スタックモード0→SPIを使用
SM=1	: スタックモード1→SPUを使用
EA=0	: SDBTを未検出
EA=1	: SDBTを検出
DB=0	: SDBTを非許可
DB=1	: SDBTを許可
IE=0	: 割り込み非許可
IE=1	: 割り込み許可
RP=0	: リピートブロック無効
RP=1	: リピートブロック有効
MD=0	: モジュロアドレッシング無効
MD=1	: モジュロアドレッシング有効
F0	: 汎用フラグ(実行コントロールフラグ)
F1	: 汎用フラグ(実行コントロールフラグ)

- F 2 : 汎用フラグ
- F 3 : 汎用フラグ
- F 4 (S) : 飽和演算フラグ
- F 5 (V) : オーバーフローフラグ
- F 6 (VA) : 累積オーバーフローフラグ
- F 7 (C) : キャリー／ボローフラグ

【0057】

以下、このマイクロプロセッサの命令一覧を示す。

A. マイクロプロセッサ機能に関する命令

A-1. ロード／ストア命令

- LDB : Load one byte to a register with sign extension
[1バイトロード (符号拡張あり)]
- LDBU : Load one byte to a register with zero extension
[1バイトロード (ゼロ拡張あり)]
- LDH : Load one half-word to a register with sign extension
[1ハーフワードロード (符号拡張あり)]
- LDHH : Load one half-word to a register high
[1ハーフワードロード (レジスタ上位へ)]
- LDHU : Load one half-word to a register with zero extension
[1ハーフワードロード (ゼロ拡張あり)]
- LDW : Load one word to a register
[1ワードロード]
- LD2W : Load two words to registers
[2ワードロード]
- LD4BH : Load four bytes to four half-words in two registers
with sign extension
[4バイトロード (2レジスタへ, 符号拡張あり)]
- LD4BHU : Load four bytes to four half-words in two registers
with zero extension

	[4 バイトロード (2 レジスタへ, ゼロ拡張あり)]
LD2H	: Load two half-words to two words in two registers with sign extension
	[2 ハーフワードロード (2 レジスタへ, 符号拡張あり)]
STB	: Store one byte from a register
	[1 バイトストア]
STH	: Store one half-word from a register
	[1 ハーフワードストア]
STHH	: Store one half-word from a register high
	[1 ハーフワードストア (レジスタ上位から)]
STW	: Store one word from a register
	[1 ワードストア]
ST2W	: Store two words from registers
	[2 ワードストア]
ST4HB	: Store four bytes from four half-words from two registers
	[4 バイトストア (2 レジスタの4 ハーフワードから)]
ST2H	: Store two half-words from two registers
	[2 ハーフワードストア (2 レジスタから)]
MODDEC	: Decrement a register value by a 5-bits immediate value
	[即値5ビットのデクリメント]
MODINC	: Increment a register value by a 5-bits immediate value
	[即値5ビットのインクリメント]

【0058】

A-2. 転送命令

MVFSYS	: Move a control register to a general purpose register
	[制御レジスタから汎用レジスタへ]
MVTSYS	: Move a general purpose register to a control register
	[汎用レジスタから制御レジスタへ]

MVFACC : Move a word from an accumulator

[アキュムレータからの1ワード転送]

MVTACC : Move two general purpose registers to an accumulator

[2汎用レジスタ内容のアキュムレータへの転送]

【0059】

A-3. 比較命令

CMPcc : Compare [比較]

cc = EQ (等しい), NE (不等), GT (より大),

GE (以上), LT (未満), LE (以下),

PS (ともに正), NG (ともに負)

CMPUcc : Compare unsigned [比較 (符号なし)]

cc = GT, GE, LT, LE

【0060】

A-4. 最大値/最小値命令

reserved [予約済]

【0061】

A-5. 算術演算命令

ABS : Absolute [絶対値をとる]

ADD : Add [加算]

ADDC : Add with carry [加算 (キャリー付き)]

ADDHppp : Add half-word [ハーフワード加算]

ppp = LLL (レジスタ下位, レジスタ下位, レジスタ下位),
LLH (レジスタ下位, レジスタ下位, レジスタ上位),
LHL, LHH, HLL, HLH, HHL, HHH

ADDS : Add register Rb with the sign of the third operand

[レジスタ Rb に符号を付ける]

ADDS2H : Add sign to two half-words

[2ハーフワードに符号を付ける]

ADD2H : Add two pairs of half-words
[2ハーフワード同士の加算]

AVG : Average with rounding towards positive infinity
[平均をとる]

AVG2H : Average two pairs of half-words rounding
towards positive infinity
[2ハーフワードそれぞれの平均をとる]

JOINpp : Join two half-words [2ハーフワードの結合]
pp = LL, LH, HL, HH

SUB : Subtract [減算]

SUBB : Subtract with borrow [ボロー付き減算]

SUBHppp : Subtract half-word [ハーフワードの減算]
ppp = LLL, LLH, LHL, LHH, HLL,
HLH, HHL, HHH

SUB2H : Subtract two pairs of half-words
[2つのハーフワードの減算]

【0062】

A-6. 論理演算命令

AND : logical AND [論理積]

OR : logical OR [論理和]

NOT : logical NOT [反転]

XOR : logical exclusive OR [排他的論理和]

ANDFG : logical AND flags [フラグの論理積]

ORFG : logical OR flags [フラグの論理和]

NOTFG : logical NOT a flag [フラグの反転]

XORFG : logical exclusive OR flags [フラグの排他的論理和]

【0063】

A-7. シフト演算命令

SRA : Shift right arithmetic [算術右シフト]

SRA2H : Shift right arithmetic two half-words
[2つのハーフワードの算術右シフト]

SRC : Shift right concatenated registers
[レジスタ連鎖右シフト]

SRL : Shift right logical [論理右シフト]

SRL2H : Shift right logical two half-words
[2つのハーフワードの論理右シフト]

ROT : Rotate right [右回転]

ROT2H : Rotate right two half-words
[2つのハーフワードの右回転]

【0064】

A-8. ビット操作命令

BCLR : Clear a bit [ビットクリア]

BNOT : Invert a bit [ビット反転]

BSET : Set a bit [ビットセット]

BTST : Test a bit [ビットテスト]

【0065】

A-9. 分岐命令

BRA : Branch [分岐]

BRA TZ R : Branch if zero [ゼロなら分岐]

BRA TN Z : Branch if not zero [ゼロでないなら分岐]

BSR : Branch to subroutine [サブルーチンへ分岐]

BSR TZ R : Branch to subroutine if zero
[ゼロならサブルーチンへ分岐]

BSR TN Z : Branch to subroutine if not zero
[ゼロでないならサブルーチンへ分岐]

JMP : Jump [無条件ジャンプ]

JMPT Z R : Jump if zero [ゼロならジャンプ]

JMPT N Z : Jump if not zero [ゼロでないならジャンプ]

J S R : Jump to subroutine [サブルーチンヘジャンプ]

J S R T Z R : Jump to subroutine if zero
[ゼロならサブルーチンヘジャンプ]

J S R T N Z : Jump to subroutine if not zero
[ゼロでないならサブルーチンヘジャンプ]

N O P : No Operation [無操作]

[ディレイド分岐, ジャンプ命令]

D B R A : Delayed branch [ディレイド分岐]

D B R A I : Delayed branch immediate [ディレイド分岐 (即値)]

D B S R : Delayed branch to subroutine
[ディレイドサブルーチン分岐]

D B S R I : Delayed branch immediate to subroutine
[ディレイドサブルーチン分岐 (即値)]

D J M P : Delayed jump [ディレイドジャンプ]

D J M P I : Delayed jump immediate [ディレイドジャンプ (即値)]

D J S R : Delayed jump to subroutine
[ディレイドサブルーチンジャンプ]

D J S R I : Delayed jump immediate to subroutine
[ディレイドサブルーチンジャンプ (即値)]

【0066】

A-10. OS 関連命令

T R A P : Trap [トラップ]

R E I T : Return from exception, interrupts and traps
[例外、割り込み、トラップからのリターン]

【0067】

B. DSP 機能に関する命令

B-1. 算術操作命令

M U L : Multiply [乗算]

M U L X : Multiply with extended precision [倍精度乗算]

- MULXS** : Multiply and shift to the right by one
with extended precision
[倍精度乗算および1ビット右シフト]
- MULX2H** : Multiply two pairs of half-words
with extended precision
[2ハーフワードずつの倍精度乗算]
- MULHX p p** : Multiply two half-words with extended precision
p p = LL, LH, HL, HH
[2ハーフワードの倍精度乗算]
- MUL2H** : Multiply two pairs of half-words
[2ハーフワードずつの乗算]
- MAC a** : Multiply and add [積和演算]
a (アキュムレータ指定) = 0, 1
- MACS a** : Multiply, shift to the right by one and add
a = 0, 1
[1ビット右シフト付き積和演算]
- MSUB a** : Multiply and subtract [積和(減算)演算]
a = 0, 1
- MSUBS a** : Multiply, shift to the right by one and subtract
a = 0, 1
[1ビット右シフト付き積和(減算)演算]

【0068】

B-2. リピート命令

- REPEAT** : Repeat a block of instructions
[命令ブロックの繰り返し]
- REPEAT I** : Repeat a block of instructions immediate
[命令ブロックの繰り返し(即値指定)]

【0069】

次に、図2の命令フォーマット内のCDレジスタについて説明する。

CDフォーマット404は、対応している演算フィールド106で指定された operation_0 のパイプラインでの処理において、対応している実行条件フィールド401で指定される実行条件を判定する時期を遅延させる遅延量を指定するもので、ユーザがCDフィールド404の値を適宜設定することによりその遅延量を可変に設定することができる。

【0070】

具体的にCDフィールド404には、演算フィールド106の記述された命令フォーマット101が格納されているメモリアドレスXからのオフセット値aが即値として記述される。この場合、operation_0 の処理において実行条件フィールド401に記述された実行条件が判断される時期は、マイクロプロセッサ1のPC値が $(X + a)$ 番地を保持しているクロックサイクルとなる。オフセット値aはゼロであってもよい。その場合実行条件が判断される時期は、PCの値がX番地を保持しているクロックサイクルとなる。さらに、CDフィールド404には、アドレス値を保持するプロセッサ1内のレジスタの番号を指定する情報を記述することもできる。その場合、実行条件フィールド401に記述された実行条件が判断される時期は、マイクロプロセッサ1のPC値がその指定されたレジスタに保持されたアドレスであるクロックサイクルとなる。CDフィールド404に即値が記述してあるかレジスタ番号が記述してあるかを識別するビットが同じCDフィールド404にある。

但し、実行条件フィールド401がCC=000の無条件実行を示している場合、対応するoperation_0 の実行処理においてCDフィールド404の値は命令デコードユニット2や各ユニット3、4で無視される。

【0071】

CDフィールド405は、演算フィールド107及び条件実行フィールド402に対して、CDフィールド404と同一である。また、CDフィールド406も、フィールド108～110の演算及び条件実行フィールド403に対して、CDフィールド404と同一である。

【0072】

図1に示されたメモリユニット3内の各レジスタについて説明する。

PC制御部13のレジスタ30、メモリ制御部14のレジスタ40、ALU15のレジスタ50、及びシフタ16のレジスタ60の各々は、演算フィールド106で指定されるoperation_0 がデコードされて得られる制御信号11をそのまま保持する。

【0073】

PC制御部13のレジスタ31、メモリ制御部14のレジスタ41、ALU15のレジスタ51、及びシフタ16のレジスタ61の各々は、operation_0 の演算が実行される条件を指定する情報を保持する。本実施の形態の場合には、命令フォーマット101内の条件実行フィールド401のCC値がそのまま保持される。

【0074】

PC制御部13のレジスタ32、メモリ制御部14のレジスタ42、ALU15のレジスタ52、及びシフタ16のレジスタ62の各々は、operation_0 の演算の実行条件を判断する時期の情報を保持する。実際にはoperation_0 の演算の実行条件を判断するときのPC値（アドレス値）が保持される。例えばPC制御部13がオフセット値を有するCDフィールド404に基づいて、そのレジスタ32に保持されるべきアドレス値を生成する。PC制御部13は命令デコードユニット2からCDフィールド404のオフセット値を受け取り、operation_0 の記述された命令フォーマット101のアドレスにそのオフセット値を加算し、その加算値をレジスタ32にセットする。CDフォーマット404がレジスタ番号を指定している場合には、PC制御部13は命令デコードユニットからそのレジスタ番号を受け取ったそのままレジスタ32にセットする。レジスタ32の保持するレジスタ番号で特定されるレジスタには、operation_0 の演算の実行条件を判断するときのPC値（アドレス値）を保持されている。なお、レジスタ32にPC値が保持されているかレジスタ番号が保持されているかを識別するビットがレジスタ32に設けられている。

メモリ制御部14、ALU15及びシフタ16は、PC制御部13と同様の方法により、PC値あるいはレジスタ番号を生成してそれぞれレジスタ42、52および62へセットする。

【0075】

PC制御部13は、演算フィールド106で指定されるoperation_0が、上記(A-9)にあるような分岐命令である場合に命令実行部の演算ユニットとして動作するものであり、operation_0が分岐命令であればレジスタ30～32が上述のように機能し、他のレジスタ40～42、50～52、60～62は使用されない。

またメモリ制御部14は、演算フィールド106で指定されるoperation_0が、上記(A-1)のロード/ストア命令のようなメモリアクセス命令である場合に命令実行部の演算ユニットとして動作するものであり、operation_0がメモリアクセス命令であればレジスタ40～42が上述のように機能し、他のレジスタは使用されない。

またALU15は、演算フィールド106で指定されるoperation_0が、上記(A-5)にあるような算術演算命令、又は(A-6)にあるような論理演算命令である場合に命令実行部の演算ユニットとして動作するものであり、operation_0が算術演算命令あるいは論理演算命令であればレジスタ40～42が上述のように機能し、他のレジスタは使用されない。

さらに、シフタ16は、演算フィールド106で指定されるoperation_0が、上記(A-7)にあるようなシフト演算命令である場合に命令実行部の演算ユニットとして動作するものであり、operation_0がシフト演算命令であればレジスタ40～42が上述のように機能し、他のレジスタは使用されない。

【0076】

PC制御部13内のレジスタ33、及び記憶回路34は、operation_0が特にディレイド分岐命令およびディレイドジャンプ命令である場合に使用されるものである。

図7はディレイド分岐命令の基本的なフォーマット320を示す説明図である。基本的には、ディレイド分岐命令のフォーマット320は、オペコード321、分岐を実行する時期を遅延させるための遅延量を指定する実行遅延量指定フィールド322および分岐先アドレスを指定するためのオフセットまたはアドレスが指定される分岐先指定フィールド323を持つ。ディレイド分岐命令は、例え

ば、図4に示すフォーマット116 (Short_D1)、フォーマット117 (Short_D2) またはフォーマット118 (Long) で実現される。フォーマット116 (Short_D1) は、遅延量としてレジスタ設定値が用いられる場合に使用される。フォーマット117 (Short_D2) は、遅延量として即値が用いられる場合に使用される。フォーマット118 (Long) は、分岐先アドレスを32ビット即値で指定する場合に使用される。各フォーマットにおいて、オペコードはフィールド120で指定される。また、フィールド129は、フィールド128がレジスタ番号を示すのか即値を示すのかを指定するために使用される。フィールド121は、DBRA, DBSR, DJMP, DJSRの各命令において遅延量がレジスタで指定されときのレジスタ指定領域として使用され、フィールド131は、遅延量を指定する即値の領域として使用される。

【0077】

ディレイドジャンプ命令も図7に示すフォーマットで記述される。但し、フィールド323では、ジャンプ先のアドレスを保持しているレジスタのレジスタ番号が指定されている。

【0078】

本実施の形態の場合、フィールド131に即値として記述された遅延量はディレイド分岐（あるいはジャンプ）命令の記述されている2演算命令101のアドレスXからのオフセット値bである。よってユーザは実行遅延量指定フィールド322の値を適宜設定することによりその遅延量は可変とすることができる。ただし分岐の実行される時期がCDフィールド404により決定される分岐の条件を判断する時期より前にならないように、ユーザは実行遅延量指定フィールド322あるいはCDフィールド404を設定しなければならない。

【0079】

PC制御部13に備えられたレジスタ33は、ディレイド分岐（あるいはジャンプ）命令で指定される分岐を実行する時期の情報を保持する。実際には分岐を実行するときのPC値（アドレス値）が保持される。PC制御部13がオフセット値を有するフィールド322に基づいて、そのレジスタ33に保持されるべき

アドレス値を生成する。PC制御部 13 は命令デコードユニット 2 からフィールド 322 のオフセット値を受け取り、ディレイド分岐（あるいはジャンプ）命令の記述された命令フォーマット 101 のアドレスにそのオフセット値を加算し、その加算値をレジスタ 33 にセットする。フォーマット 322 がレジスタ番号を指定している場合には、PC制御部 13 は命令デコードユニットからそのレジスタ番号を受け取ったそのままレジスタ 33 にセットする。レジスタ 33 の保持するレジスタ番号で特定されるレジスタには、分岐を実行するときの PC 値（アドレス値）を保持されている。

【0080】

また PC 制御部 13 の記憶回路 34 は、PC 制御部 13 がディレイド分岐（あるいはジャンプ）命令の実行条件を判断した結果、実行条件を満たしたか満たしていないかを示す情報を保持するものである。

【0081】

なお、レジスタ 30 に保持されている制御信号 11 には、その制御信号 11 に対する命令が、ディレイド分岐命令であるかディレイなしの通常の分岐命令であるかを識別する 1 ビットの情報を含んでいる。

【0082】

さらに、図 1 に示された整数演算ユニット 3 内の各レジスタについて説明する。

乗算器 17 のレジスタ 70、ALU 19 のレジスタ 80、及びシフタ 20 のレジスタ 90 の各々は、演算フィールド 106 で指定される operation_1 がデコードされて得られる制御信号 12 をそのまま保持する。

【0083】

乗算器 17 のレジスタ 71、ALU 19 のレジスタ 81、及びシフタ 20 のレジスタ 91 の各々は、operation_1 の演算が実行される条件を指定する情報を保持する。本実施の形態の場合には、命令フォーマット 101 内の条件実行フィールド 402 の CC 値がそのまま保持される。

【0084】

乗算器 17 のレジスタ 72、ALU 19 のレジスタ 82、及びシフタ 20 のレ

レジスタ 92 の各々は、operation_1 の演算の実行条件を判断する時期の情報を保持する。CD フィールド 405 がオフセット値であれば、operation_1 の演算の実行条件を判断するときの PC 値（アドレス値）として各レジスタに保持される。その保持されるべき PC 値は operation_1 の記述された命令フォーマット 101 のアドレスとそのオフセット値との加算値である。また CD フィールド 405 がレジスタ番号を指定しているならば、そのレジスタ番号がそのまま保持される。レジスタ 72 の保持するレジスタ番号で特定されるレジスタには、operation_0 の演算の実行条件を判断するときの PC 値（アドレス値）を保持されている。レジスタ 72、82 及び 92 への値のセットは、CD フィールド 405 を命令デコードユニットから受け取る CD フィールド 405 に従い、それぞれ乗算器 17、ALU 19 及びシフタ 20 により行われる。

【0085】

乗算器 17 は、演算フィールド 107 で指定される operation_1 が上記（B-1）にあるように乗算を伴う乗算命令および積和命令である場合に命令実行部の演算ユニットとして動作するものであり、operation_1 が乗算を伴う命令であればレジスタ 70～72 が上述のように機能し、他のレジスタ 80～82、90～92 は使用されない。

また ALU 19 は、演算フィールド 107 で指定される operation_1 が、上記（A-5）にあるような算術演算命令、又は（A-6）にあるような論理演算命令である場合に命令実行部の演算ユニットとして動作するものであり、operation_1 が算術演算命令あるいは論理演算命令であればレジスタ 80～82 が上述のように機能し、他のレジスタは使用されない。

さらに、シフタ 20 は、演算フィールド 107 で指定される operation_1 が、上記（A-7）にあるようなシフト演算命令である場合に命令実行部の演算ユニットとして動作するものであり、operation_1 がシフト演算命令であればレジスタ 40～42 が上述のように機能し、他のレジスタは使用されない。

【0086】

図 2（b）に示す命令フォーマット 102 による 1 演算命令が処理される場合には、分岐、メモリアクセス、算術演算等の種類によって、PC 制御部 13、メ

メモリ制御部 14、ALU 15、19、シフタ 16、20 及び乗算器 17 のうちの 1 つが命令実行部の演算ユニットとして演算命令を実行する。CD フィールド 406 は、同様に 1 演算命令のパイプラインでの処理において、実行条件フィールド 403 で指定される実行条件を判定する時期を遅延させる遅延量を指定する。そして 1 演算命令 102 を実行する命令実行部の演算ユニットに備えられたレジスタは、その 1 演算命令 102 について上述したものと同一内容の値を保持する。

【0087】

次に、図 8 に示されたプログラム例を用いてマイクロプロセッサ 1 の動作を説明する。

このプログラムにおいて、各行における 2 つのサブ命令が、図 2 (a) の命令フォーマットの 2 演算命令 101 で記述され、サブ命令 I01、I11、I21、I31、I41、I51 及び I61 が operation_0 として演算フィールド 106 で記述され、サブ命令 I02、I12、I22、I32、I41、I51 及び I61 が operation_1 として演算フィールド 107 で記述されている。それぞれの 2 演算命令は、同じ行に記述されたメモリのアドレス番地によりアクセスされる。例えば、サブ命令 I01、I02 は 1000～1007 番地のメモリの領域に格納され、1000 番地のアドレスでアクセスされる。

【0088】

サブ命令 I01 は、「フラグ F0 が偽（つまりフラグが 0 を保持している）」という分岐条件を満たしていれば、loop の記述されたサブ命令 I11、I12 に分岐する分岐命令 BRA で、この分岐条件を end の記述されたサブ命令 I41、I42 の実行時に判断することのできる。また分岐命令 BRA はディレイなしの通常分岐命令である。サブ命令 I21 は、レジスタ R2 とレジスタ R3 とのそれぞれ内容を加算してレジスタ R2 に格納する加算命令 ADD である。サブ命令 I31 は、レジスタ R2 とレジスタ R4 との内容が等しければフラグ F0 に“1”を書き込み、等しくなければ“0”を書き込む比較命令 CMPEQ である。サブ命令 I11、I41、I51、I61 は、メモリユニット 2 でステージ E/M の処理がなされる任意の算術演算命令で、その他のサブ命令は整数演算ユニ

ット3でステージE/Mの処理がなされる任意の算術演算命令である。

【0089】

サブ命令I01に対応する実行条件フィールド401には、“CC=010”が記述されており、サブ命令I01以外のすべてのサブ命令は無条件に実行する命令であるとし、その対応の実行条件フィールドには“CC=000”が記述されている。

また、サブ命令I01に対応するCDフィールド404にはオフセット値’20’が記述されている。

また、各行に記述された2演算命令は、FMフィールド103、104が00に設定されて2つのサブ命令の演算を並列に実行する命令であるとする。

【0090】

図8のプログラムにおいて、1008番地、1010番地、1018番地、1020番地の4つの2演算命令で1つのループが形成されており、このプログラムはサブ命令I31の比較命令の実行にフラグF0が真になるまでこのループ内の4つの2演算命令が繰り返して順次実行されることを意味している。

【0091】

図9は、図8に示されたプログラムをパイプライン処理するマイクロプロセッサ1の動作を示している。図においてクロックt1ないしt13は連続する1クロックサイクルを示し、各クロックサイクル内のすべてのパイプラインステージの処理は並列に行われる。例えば、クロックt5において、サブ命令I11、I12の各ステージW、サブ命令I21、I22の各ステージE/M、サブ命令I31、I32の各ステージD/A、サブ命令I41、I42の各ステージIFは、すべて並列に処理される。他のクロックサイクルに関しても同様である。

【0092】

図9において、マイクロプロセッサ1のPCが保持するアドレス値は、そのアドレス値を有するクロックサイクルが、そのアドレス値によりアクセスされる2演算命令が命令デコードステージD/Aで処理されるサイクルの次サイクルであることを示す。

【0093】

サブ命令 I 0 1、I 0 2 に関し、命令フェッチステージ IF、命令デコードステージ D/A はそれぞれクロック t 1、t 2 で並列に処理される。サブ命令 I 0 2 の命令実行ステージ E/M 及びライトバックステージ W はそれぞれクロック t 3、t 4 で処理される一方、サブ命令 I 0 1 の命令実行ステージ E/M は許可されるまで実行条件の判断、およびそれに基づく分岐の実行はなされない。

【0094】

サブ命令 I 0 1 である分岐命令 B R A の条件指定フィールド 4 0 1 と C D フィールド 4 0 4 とに従って、命令デコードユニット 2 はサブ命令 I 0 1 が実行条件の判断を遅延させる命令であることを検出し、この条件指定フィールドと C D フィールドの内容が実行条件を遅延して判断するための制御信号として P C 制御部 1 3 へ出力される。また演算指定フィールドがデコーダ 8 でデコードされて、このデコード結果に応じた制御信号 1 1 が出力される。この制御信号 1 1 には、分岐命令 B R A を生じさせるように P C 制御部 1 3 を制御するための第 1 の情報、サブ命令 I 0 1 が実行の遅延を生じさせない通常の命令であることを示す第 2 の情報、さらに分岐命令 B R A の分岐先アドレスを示す第 3 の情報を含んでいる。この第 3 の情報は分岐先アドレスそのものであり、分岐命令 B R A のフィールド 3 2 3 で指定されたオフセットに基づき命令デコードステージ E/A でアドレス計算専用の加算器（図示していない）で計算されたものである。

【0095】

クロック t 3 において、P C 制御部 1 3 は、命令デコードユニット 2 からサブ命令 I 0 1 が条件付きの命令であり、その条件を判断する時期を遅延させる命令であることの情報を受けて分岐命令 B R A に対する制御信号 1 1 をレジスタ 3 0 にそのまま保持する。まだ制御信号 1 1 に基づいて分岐は実行されない。命令デコードユニット 2 から出力された実行条件フィールドである c c = 0 1 0 の値がレジスタ 3 1 にそのまま保持される。さらに P C 制御部 1 3 は、クロック t 3 において命令デコードユニット 2 から C D フィールドであるオフセット値 “2 0” を、P C から 1 0 0 0 番地をそれぞれ受け取って加算し、レジスタ 3 2 がその加算結果である 1 0 2 0 番地を保持する。P C 制御部 1 3 はレジスタ 3 2 に保持さ

れた値とPCの示す値とを比較している。PCのアドレス値がレジスタ32のアドレス値と等しいクロックサイクル、つまりクロック t_7 でレジスタ31に保持されたcc値に基づきPC制御部13が分岐命令BRAの実行条件を初めて判断する。

【0096】

一方サブ命令I02に関し、命令デコードユニット2のデコーダ9が演算指定フィールド107を解析して、ALU18に算術演算を行わせるための制御信号12が出力される。そして命令デコードユニット2はこのサブ命令I01が条件指定フィールドに基づきサブ命令I02が無条件命令であることを検知し無条件(cc=000)であることを示す情報も出力する。

ALU18は無条件であることを示す情報を受けて、制御信号12をレジスタ80に保持することなく制御信号12に従って加算演算を行う。また命令デコードユニット2からサブ命令I01における実行条件フィールド402の値及びCDフィールド405の値が出力されるが、ALU18はこれらの値を無視してレジスタ81、82にはそれまで保持していた値をそのまま保持させる。以下に示す他の無条件実行のサブ命令についても同様である。

【0097】

サブ命令I11、I12に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M及びライトバックステージWは、それぞれクロック t_2 、 t_3 、 t_4 、 t_5 で並列に処理される。

【0098】

サブ命令I21、I22に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M及びライトバックステージWは、それぞれクロック t_3 、 t_4 、 t_5 、 t_6 で並列に処理される。サブ命令I21である加算命令ADDにおいては、クロック t_5 のステージE/MでレジスタR5とレジスタR6との内容がALU15により加算演算され、クロック t_6 のステージWでその加算結果がレジスタR5に書き込まれる。

【0099】

サブ命令I31、I32に関し、命令フェッチステージIF、命令デコードステ

ージD/A、命令実行ステージE/Mは、それぞれクロックt4、t5、t6で並列に処理される。サブ命令I31である比較命令CMP EQにおいては、クロックt6のステージE/Mで、レジスタR2とレジスタR4との内容がALU15により比較され、等しければフラグF0に‘1’を書き込み、等しくなければフラグF0に‘0’を書き込む。比較命令CMP EQの実行ステージE/Mは、本来加算命令ADDによる演算結果がレジスタr5に書き込まれるクロックt6まで開始できないが、プロセッサ1内部に設けられたバイパス回路によってクロックt5で得られる加算命令ADDの演算結果を用いて処理される。

なお、比較命令CMP EQの場合、ライトバックステージWは存在しない。一方サブ命令I32のライトバックステージWはクロックt7で処理される。

【0100】

サブ命令I41、I42に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M及びライトバックステージWは、それぞれクロックt5、t6、t7、t8で並列に処理される。

【0101】

分岐命令BRAの分岐条件は、サブ命令I31の演算結果を参照しなければならないためクロックt7で判定される。クロックt7において、PC制御部13はレジスタ31に保持された‘010’の値に従ってフラグF0を参照し、フラグF0が‘0’ならば分岐を発生させることを決定し、‘1’ならば分岐を発生させないことを決定する。

【0102】

PC制御部13はレジスタ30に保持されている制御信号11の第2の情報が分岐命令BRAがディレイなしの分岐命令であることを示していることに従い、レジスタ33の内容を無視する。つまり、PC制御13において、分岐条件が決定されたことに従い、同じクロックt7で命令実行ステージE/Mが処理される。命令I41は算術演算命令であり、PC制御部13ではなくALU15で命令実行ステージE/Mの処理がなされているので、サブ命令I01、I41のステージE/Mは並列に行える。

【0103】

図9に示されたマイクロプロセッサ1の動作において、比較命令CMP EQでフラグF0に‘0’が書き込まれとすると、PC制御部13が実行条件を満たしたことを示す情報（例えば‘1’の論理値）を生成することにより条件判断が完了され、その情報に応答してレジスタ30に保持された制御信号11に基づき分岐が実行される。具体的には、制御信号11の第3の情報に従いPC制御部13は分岐先アドレス（この場合1008番地）をIAバスを介して命令RAM6へ出力するとともに、その分岐先アドレスを次クロックサイクルでPCへセットする。さらに制御信号11の第1の情報に従い、PC制御部13は分岐先アドレス1008に格納されている2演算命令（サブ命令I11、I21）を命令デコードユニット2へ与えるように命令RAM6を制御する。PC制御部13はこの第1の情報に従い、クロックt7ですでにステージIF、E/Aが処理されているサブ命令I51、I52、及びすでにステージIFの処理がされているサブ命令I61、I62のパイプラインの処理をキャンセルするように命令デコードユニット2、メモリユニット3および整数演算ユニット4を制御する。但し、クロックt7でステージE/Mの処理がなされたサブ命令I41、I42のライトバックステージWは、キャンセルされずに処理される。

【0104】

命令デコードユニット2は、命令RAM6から出力された1008番地にあるサブ命令I11、I12の2演算命令を受け取り、クロックt8で命令フェッチステージIFが処理されたことになる。サブ命令I11、I12の命令デコードステージD/A、命令実行ステージE/M、ライトバックステージWがそれぞれクロックt9、t10、t11で処理される。PCは、1008番地の2演算命令における命令実行ステージE/Mが処理されるクロックt10まで1008番地をそのまま保持し、クロックt10以降8ずつカウントする。

【0105】

サブ命令I11、I12の2演算命令に続き、1010番地、1018番地、1020番地のそれぞれ2演算命令は1クロックずつ遅れて順にパイプラインにて処理される。そしてPC制御部13のレジスタ30、31、32は、クロック

t 3 で保持した値をそのまま保持しているので、PC が 1020 番地を保持するクロック t 13 において、レジスタ 30、31、32 の内容に基づき、PC 制御 13 がフラグ F0 を参照して「フラグ F0 の値が '0' である」という同じ実行条件を再び判断して分岐命令 BRA の分岐を実行する。図示していないが実行条件は、クロック t 12 において得られるサブ命令 I 31 の実行結果であるフラグ F0 の更新で決定される。

【0106】

クロック t 7 において、分岐命令 BRA の実行条件を満たされていないと判断された場合には、PC 制御部 13 において実行条件を満たしていないことを示す情報（例えば、'0' の論理値）が生成され、その情報に応答してレジスタ 30 に保持された制御信号 11 を参照することなく分岐は実行されない。サブ命令 I 51、I 52 の 2 演算命令、サブ命令 I 61、I 62 の 2 演算命令のそれぞれパイプラインステージの処理がキャンセルされることなく引き続き行われる。また、レジスタ 30、31、32 に保持されている内容は、次の条件付き分岐命令により更新されるまでそのまま残しておいてもよいし、すべてリセットしてもよい。

【0107】

また、分岐命令 BRA に対する CD フィールド 404 のオフセット値がゼロであれば、PC 値が分岐命令自身のアドレスである 1000 番地を示すクロック t 3 で実行条件が判断され、分岐が実行される。

【0108】

また、図 10 のプログラム例を用いてマイクロプロセッサ 1 の別の動作を説明する。

図 8 と同じく、各行における 2 つのサブ命令が、図 2 (a) の命令フォーマットの 2 演算命令 101 で記述され、サブ命令 I 01、I 11、I 21、I 31、I 41、I 51 及び I 61 が operation_0 として演算フィールド 106 で記述され、サブ命令 I 02、I 12、I 22、I 32、I 41、I 51 及び I 61 が operation_1 として演算フィールド 107 で記述されている。

【0109】

サブ命令 I 0 1 は、「フラグ F 0 が真（‘1’を保持している）」という実行条件を満たしていれば、レジスタ R 5 に保持されているアドレス番地の命令にジャンプするジャンプ命令 D J M P であり、ジャンプ命令 D J M P は、図 7 に示されたフォーマットのディレイドジャンプ命令である。サブ命令 I 1 1 は、レジスタ R 1 とレジスタ r 2 との内容を加算して加算結果をレジスタ R 1 へ書き込む加算命令 A D D である。サブ命令 I 2 1 は、レジスタ R 1 とレジスタ R 3 との内容を比較し、等しければフラグ F 0 に‘1’を書き込み、等しくなければフラグ F 0 に‘0’を書き込む比較命令 C M P E Q である。サブ命令 I 3 1 は、レジスタ R 5 とレジスタ R 6 との内容を加算して加算結果をレジスタ R 5 に書き込む加算命令 A D D である。サブ命令 I 1 1、I 4 1、I 5 1、I 6 1 は、メモリユニット 2 でステージ E/M の処理がなされる任意の算術演算命令で、その他のサブ命令は整数演算ユニット 3 でステージ E/M の処理がなされる任意の算術演算命令である。

【0110】

サブ命令 I 0 1 に対応する実行条件フィールド 4 0 1 には、“C C = 0 0 1”が記述されており、サブ命令 I 0 1 以外のすべてのサブ命令は無条件に実行する命令であるとする。

また、サブ命令 I 0 1 に対応する C D フィールド 4 0 4 にはオフセット値“20”が記述されている。

さらに、サブ命令 I 0 1 の実行遅延量指定フィールド 3 2 2（図 7）には即値としてオフセット値‘28’が記述されている。

また、各行に記述された 2 演算命令は、F M フィールド 1 0 3、1 0 4 が 0 0 に設定されて 2 つのサブ命令の演算を並列に実行する命令であるとする。

【0111】

図 10 のプログラムにおいて、サブ命令 I 0 1 のジャンプ命令 D J M P によるジャンプは、サブ命令 I 2 1 の比較結果であるフラグ F 0 の値を判断して実行され、且つそのジャンプ先命令のアドレスはサブ命令 I 3 1 の加算結果で決定される。そして、ジャンプ命令 D J M P によりジャンプするしないに拘わらず 1 0 2

0番地、1028番地の2演算命令は実行されるものである。

【0112】

図11は、図8に示されたプログラムをパイプライン処理するマイクロプロセッサ1の動作を示している。図9と同じく、クロック t_1 ないし t_13 は連続する1クロックサイクルを示し、各クロックサイクル内のすべてのパイプラインステージの処理は並列に行われる。PCの値の意味も図9と同じである。

【0113】

サブ命令I01、I02に関し、命令フェッチステージIF、命令デコードステージD/Aはそれぞれクロック t_1 、 t_2 で並列に処理される。サブ命令I02の命令実行ステージE/M及びライトバックステージWはそれぞれクロック t_3 、 t_4 で処理される一方、サブ命令I01の命令実行ステージE/Mは許可されるまで実行条件の判断、およびそれに基づく分岐の実行はなされない。

【0114】

サブ命令I01であるディレイドジャンプ命令DJMPは図1におけるデコーダ8でデコードされて、このデコード結果に応じた制御信号11が出力される。この制御信号11には、ジャンプ命令BRAを生じさせるようにPC制御部13を制御するための第1の情報、サブ命令I01が実行の遅延を生じさせる命令であることを示す第2の情報、さらに分岐命令BRAの分岐先アドレスを示す第3の情報を含んでいる。第3の情報はジャンプ命令DJMPで指定されているレジスタ番号である。

【0115】

クロック t_3 において、PC制御部13はジャンプ命令DJMPに対する制御信号11をレジスタ30にそのまま保持し、その制御信号11に基づき分岐を実行しない。命令デコードユニット2から出力されたジャンプ命令BRAの実行条件フィールドである $cc=001$ の値がレジスタ31にそのまま保持される。さらにPC制御部13は、クロック t_3 において命令デコードユニット2からジャンプ命令DJMPの実行遅延量指定フィールド322のオフセット値“28”を、PCから1000番地をそれぞれ受け取って加算し、レジスタ32がその加算結果である1018番地を保持する。PC制御部13はレジスタ32に保持され

た値とPCの示す値とを比較している。PCのアドレス値がレジスタ32のアドレス値と等しいクロックサイクル、つまりクロックt6でレジスタ31に保持されたcc値に基づきPC制御部13が分岐命令BRAの実行条件を初めて判断する。

【0116】

さらにPC制御部13は、クロックt3において命令デコードユニット2からジャンプ命令BRAのCDフィールドであるオフセット値“18”を、PCから1000番地をそれぞれ受け取って加算し、レジスタ33がその加算結果である1028番地を保持する。PC制御部13はレジスタ33に保持された値とPCの示す値とを比較している。PCのアドレス値がレジスタ33のアドレス値と等しいクロックサイクル、つまりクロックt8でジャンプ命令DJMPによるジャンプが初めて実行される。

【0117】

サブ命令I11、I12に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M及びライトバックステージWは、それぞれクロックt2、t3、t4、t5で並列に処理される。サブ命令I11の加算命令ADDにあっては、ステージE/MでレジスタR1とレジスタR2との内容がALU15により加算され、ステージWでその加算結果がレジスタR1に書き込まれる。

【0118】

サブ命令I21、I22に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/Mは、それぞれクロックt3、t4、t5で並列に処理される。サブ命令I21である比較命令CMP EQにおいては、クロックt5の実行ステージE/Mで、レジスタR1とレジスタR3との内容がALU15により比較され、等しければフラグF0に‘1’を書き込み、等しくなければフラグF0に‘0’を書き込む。

なお、比較命令CMP EQの場合、ライトバックステージWは存在しない。一方サブ命令I32のライトバックステージWはクロックt7で処理される。

【0119】

ジャンプ命令DJMPの実行条件はサブ命令I21の演算結果であるフラグF

0を参照しなければならないためクロック t 6で判定される。クロック t 6において、PC制御部 13はレジスタ 31に保持された‘001’の値に従ってフラグ F0を参照し、フラグ F0が‘1’ならば分岐を発生させることを決定し、‘0’ならば分岐を発生させないことを決定する。そしてクロック t 6で記憶回路 34に分岐を発生させるか否かを決定した1ビットの情報をセットする。しかし実際にジャンプ命令によるジャンプはクロック t 8になるまで実行されない。

【0120】

ここで、比較命令CMP EQによりフラグ F0に‘1’が書き込まれ、実行条件が満たされた場合を考える。

記憶回路 34には実行条件が満たされたという情報である‘1’がセットされる。レジスタ 30に保持されている制御信号 11のジャンプ命令DJMPがディレイのあるジャンプ命令であることを示している第2の情報に従い、PC制御部 13はレジスタ 33に保持されているアドレス値がまだPCの値と一致しているかどうかをチェックする。実行条件を満たしていると決定したとしてもクロック t 6の段階ではまだ一致していないため、レジスタ 30に保持されている制御信号 11の第1および第3の情報に従ってPC制御 13は分岐を実行しない。

【0121】

サブ命令 I 31、I 32に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M及びライトバックステージWは、それぞれクロック t 4、t 5、t 6、t 7で並列に処理される。サブ命令 I 31の加算命令ADDにあっては、ステージE/Mでレジスタ R5とレジスタ R6との内容がALU 15により加算され、ステージWでその加算結果がレジスタ R5に書き込まれる。

【0122】

サブ命令 I 41、I 42に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M及びライトバックステージWは、それぞれクロック t 5、t 6、t 7、t 8で並列に処理される。また、サブ命令 I 51、I 52に関し、命令フェッチステージIF、命令デコードステージD/A、命令実行ステージE/M及びライトバックステージWは、それぞれクロック t 6、t 7、t 8、t 9で並列に処理される。

【0123】

ジャンプ命令D JMPによるジャンプはサブ命令I 31の演算結果であるレジスタR 5を参照しなければならないためクロックt 8で実行される。クロックt 8においてPC制御部13はレジスタ33に保持されているアドレス値がPCの値と一致していることを検知する。そして記憶回路34に‘1’が保持されていることに従って、PC制御部13はクロックt 8でジャンプ命令D JMPの命令実行ステージE/Mを処理する。PC制御部13はレジスタ30に保持された制御信号11に基づいて分岐を発生させる。命令I 51は算術演算命令であり、PC制御部13ではなくALU15で命令実行ステージE/Mの処理がなされているので、サブ命令I 01、I 51のステージE/Mは並列に行える。

【0124】

具体的には、制御信号11の第3の情報に従い、PC制御部13は第3の情報で指定されているレジスタに保持されている分岐先アドレス（例えば2000番地とする）をIAバスを介して命令RAM6へ出力するとともに、その分岐先アドレスを次クロックサイクルでPCへセットする。さらに制御信号11の第1の情報に従い、PC制御部13は分岐先アドレス2000に格納されている2演算命令（サブ命令I 11、I 21）を命令デコードユニット2へ与えるように命令RAM6を制御する。さらに、PC制御部13はこの第1の情報に従いクロックt 8ですでにステージIF、E/Aが処理されているサブ命令I 61、I 62、及びすでにステージIFの処理がされているサブ命令I 71、I 72のパイプラインの処理をキャンセルするように命令デコードユニット2、メモリユニット3および整数演算ユニット4を制御する。但し、クロックt 8でステージE/Mの処理がなされたサブ命令I 51、I 52のライトバックステージWは、キャンセルされずに処理される。

【0125】

命令デコードユニット2は、命令RAM6から出力された2000番地にある2演算命令を受け取り、クロックt 9でその2演算命令のそれぞれ命令フェッチステージIFが処理され、1クロックサイクル毎にステージD/A、E/M、Wが順に処理される。

【0126】

また、クロック t 6 でジャンプ命令の実行条件が満たされていない場合は、クロック t 6 で記憶回路 34 にジャンプ命令の実行条件が満たされていないという情報である '0' が保持される。そして、クロック t 8 でレジスタ 33 に保持されたアドレス値が PC の値と等しくなったとしても、記憶回路 34 に '0' が保持されていることに従い、PC 制御部 13 はレジスタ 30 の保持された制御信号 11 を参照することなく分岐を生じさせない。サブ命令 I 61、I 62 の 2 演算命令、サブ命令 I 71、I 72 の 2 演算命令のそれぞれパイプラインステージの処理がキャンセルされることなく引き続き行われる。また、レジスタ 30、31、32、33 及び記憶回路 34 に保持されている内容は、次の条件付き分岐命令により更新されるまでそのまま残してもよいし、すべてリセットしてもよい。

【0127】

また別の方法として、クロック t 6 で条件を判断し、条件を満たしていないことを示す情報に従い、レジスタ 30 に保持された制御信号 11 を、ジャンプを無効にする、つまりジャンプを実行しないように PC 制御 13 が制御される制御信号に書き換えるようにしてもよい。そして、条件を満たしているを示す情報が生成されれば、レジスタ 30 の制御信号 11 はそのまま保持されるようにする。そうすると、クロック t 8 で実際にジャンプを実行する際に参照する記憶回路 34 を設けなくてよい。

【0128】

無論、ディレイドジャンプ命令 DJMP（あるいはディレイド分岐命令 DBRA）は実行遅延量指定フィールド 322 の値を変えて、実行条件を判断するクロックサイクルと同じサイクルで分岐を実行させることもできる。

【0129】

このように実行条件を判断する時期を遅延させるのは、条件付きの分岐あるいはジャンプ命令に限るものではない。CD フォーマット 404～406 により、2 演算命令 101 の 2 つの任意のサブ命令、及び任意の 1 演算命令 102 の実行処理において、条件指定フィールド 401～403 で指定された実行条件の判断する時期を遅延させることができる。それは、PC 制御部 13 のレジスタ 30～

32と同じ構成、機能を有するそれぞれメモリ制御部14のレジスタ40～42、ALU15のレジスタ50～52、シフタ16のレジスタ60～62、乗算器17のレジスタ70～72、ALU19のレジスタ80～82、及びシフタ20のレジスタ90～92により実現される。

【0130】

例えば、図8のプログラムにおいて、サブ命令I12が

I12 (ADD F1T R13 R8,R8,R9) (A)

であるとする。つまり「フラグF1が真である」という実行条件を満たすときレジスタR8とレジスタR9との内容を加算し、その加算結果をレジスタR8に書き込む加算命令で、実行条件の判断は、PC値がレジスタR13に保持されたアドレス値となるクロックサイクルで行われる。条件指定フィールド402はcc=011、CDフィールド405はレジスタ番号‘13’がそれぞれ記述されている。レジスタ13には1018番地が保持されているものとする。

【0131】

図9において、サブ命令I12の命令実行ステージE/Mはクロックt4で処理されているが、サブ命令I12が上記(A)のような加算命令であれば、命令実行ステージE/Mはクロックt6される。

【0132】

サブ命令I12は、クロックt3でデコーダ9でデコードされ、このデコード結果に応じた制御信号12が出力される。この制御信号12には、加算命令を生じさせるようにALU19を制御するための第1の情報、演算に使用されるレジスタR8、R9を指定する第2の情報を含んでいる。

【0133】

クロックt4において、ALU19はその制御信号12をレジスタ80にそのまま保持するだけで、まだ加算を実行しない。命令デコードユニット2から出力された実行条件フィールドであるcc=011の値がレジスタ81にそのまま保持される。さらにALU19は、クロックt4において命令デコードユニット2からCDフィールドであるレジスタ番号‘13’をそのまま保持する。ALU19はレジスタ82に保持された番号‘13’のレジスタに保持された値とPC

の示す値とを比較している。PCのアドレス値が1018番地と等しいクロックサイクル、つまりクロックt6でレジスタ81に保持されたcc値に基づきALU19が加算命令の実行条件を初めて判断する。ALU19は実行条件を満たしたかどうかを示す情報を生成する。

【0134】

ALU19は、その情報に従って、条件を満たしていればレジスタ80に保持された制御信号12に基づいて同じクロックt6で加算演算を行い、クロックt7で加算結果を指定された汎用レジスタ5に保持させる。また条件を満たしていなければ、制御信号12を無視して加算演算を行わない。

また、条件を満たしていない場合の別の手段として、ALU19が制御信号12に基づき加算演算を行ってもよいが、実行条件を満たしたかどうかを示す情報に従い加算結果を加算演算命令で指定されたレジスタへ書き込まないようにALU19を構成してもよい。

【0135】

(A)の加算命令の実行条件であるフラグF1の値を決定する演算命令(2つの値を比較して、等不等あるいは大小をフラグF1に反映させる比較命令)は、プログラムシーケンス中のサブ命令I12より下位のアドレスにある命令、例えば、サブ命令I22の位置に置くことができる。

【0136】

また、CDフィールド404、405及び406の各々において、オフセット値の情報又はレジスタを指定する情報以外に、各実行条件指定フィールドで指定された実行条件が、各演算フィールドに記述された演算命令の命令デコードステージD/Aで判断されることを示す情報を記述することができる。

【0137】

例えば図8において、サブ命令I01の分岐命令BRAに対応するCDフィールド404が命令デコードステージD/Aで判断されることを示す情報を有している場合、命令デコードユニット2は、クロックt2において、CDフィールド404に応答して実行条件の判断を開始し、条件指定フィールド401に従いPSW10内の指定されたフラグF0の内容を参照する。そして命令デコードユニッ

ト 2 はサブ命令 I 0 1 が実行条件が満たされたか否かを示す第 4 の情報を出力する。また命令デコードユニット 2 はデコード 8 による分岐命令 B R A の演算指定フィールドのデコード結果に基づき制御信号 1 1 を作成する。この制御信号 1 1 には分岐命令 B R A を生じさせるように P C 制御部 1 3 を制御するための第 1 の情報、分岐実行の遅延を生じさせない通常の命令であることを示す第 2 の情報、さらに分岐命令 B R A の分岐先アドレスを示す第 3 の情報を含んでいる。

【 0 1 3 8 】

実行条件が満たしていると判断された場合には、命令デコードユニット 2 は第 4 の情報として実行条件が満たしている情報、つまり、P C 制御部 1 3 で無条件で実行してよいことを示す情報を出力する。P C 制御部 1 3 は無条件であることを示す情報を受けて、制御信号 1 1 をレジスタ 3 0 に保持することなくクロック t 3 で制御信号 1 1 に従って分岐を行う。

実行条件が満たされていないと判断された場合には、命令デコードユニット 2 が第 4 の情報として、演算が無効であることを示す情報を出力する。P C 制御部 1 3 は、演算が無効であることを示す情報に従い制御信号 1 1 を受けていても分岐を実行しない。

分岐命令以外の他の種類の演算命令についても全く同様である。ただし、加算などの算術演算命令では、命令デコードステージ D/A で演算が無効と判断された場合には、演算ユニットが算術演算を実行しても演算命令で指定されたレジスタに演算結果を書き込まないようにしてもよい。

【 0 1 3 9 】

また、図 1 に示されたこのマイクロプロセッサ 1 に、ディレイ機能を有する分岐命令 D B R A やジャンプ命令 D J M P の命令が用意されないならば、P C 制御部 1 3 にレジスタ 3 3 及び記憶回路 3 4 を設ける必要はない。

【 0 1 4 0 】

また、演算命令の命令実行部の演算ユニットとなる P C 制御部 1 3、メモリ制御部 1 4、A L U 1 5、1 9、シフタ 1 6、2 0、及び乗算器 1 7 の各々に実行条件を判断する時期を遅延させる手段であるレジスタを備えたが、これらの命令実行部の演算ユニットのうちの選択された一部にだけ備えても良い。

例えば、所定の条件の下で実行されるような演算命令のうち、ユーザの作成するプログラムで最もよく使用されるのは、条件付き分岐（あるいはジャンプ）命令である。従って、分岐命令のみの処理において、実行条件を判断する時期を遅延させる機能を持たせても良い。そうすると、PC制御部13のみにレジスタ30、31、32を備えればよい。

【0141】

以上のように、この実施の形態には次の特徴がある。

(1) 例えば、図8のサブ命令I31のような無条件実行の演算命令、あるいは条件ありでもそのデコードステージで条件が決定している演算命令であれば、クロックt5の期間で命令をデコードし、クロックt5の期間に続くクロックy6の期間で上記の条件付き演算命令の遅延量以内であるクロックt6の期間でサブ命令I31で指定された演算が実行されている。一方、サブ命令I01のような条件付き演算命令はクロックt2の期間でデコードされ、クロックt2からサブ命令I31のように命令実行ステージE/Mの処理期間であるクロックt6の時間より少なくとも長い期間が経過した後のクロックt7で分岐の演算の実行条件の判断を開始している。つまり、このクロックt3からクロックt6までの期間を遅延量として実行条件を判断する時期を遅延させている。従って、図8のプログラムシーケンスのように、条件付きの演算命令はその演算命令の実行条件を決める演算を行う演算命令（サブ命令I31）より前に置くことができる。従来のプロセッサならば、サブ命令I01の位置にNOP命令以外置くことができない可能性があるが、本実施の形態のプロセッサを採用すれば、サブ命令I01として条件付きの演算命令を置くことができ、命令のスケジューリングを柔軟にする。

【0142】

(2) 例えば各演算ユニットのレジスタ32、42等で保持される演算の実行条件を判断する時期を示す遅延量を可変に設定することができる。これにより、例えばサブ命令I01において実行条件は、クロックt4をはじめ、種々のクロックサイクルで判断されるようになる。よって、命令をスケジューリングする際、プログラムの内容に応じて条件付き演算命令の位置を適宜変えることができる。特に、レジスタ32、43等に保持される値は、プログラムを構成する命令フ

フォーマットのCDフィールド403、404。405に記述された内容に従い設定されるため、プログラマー（あるいはコンパイラ）は、その条件付き演算命令に対応させてどの程度の遅延量を要するのかを容易に決めることができる。

【0143】

(3) レジスタ32、42等で保持される遅延量としての情報は、種々の形式が考えられる。遅延量としてクロック数を保持しておいてもよい。そのときレジスタ32、42等を減算カウンタとして構成する。例えばPC制御部13において、クロックt3でクロック数「4」をレジスタ32に保持し、カウンタによりクロック数とともに減算し、「0」を保持したクロックt7で条件の判断を開始するようにすればよい。しかし、この実施の形態では、演算の実行条件を判断する時期を示す遅延量としてアドレス値が保持され、そのアドレス値プログラムカウンタの値と等しいクロックサイクルで実行条件を判断するようにしている。つまり、遅延量をPCの値で管理しているためより効果的である。例えば図8のプログラムにおいて1008番地と1020番地との間で他の命令にジャンプし、復帰する場合でもレジスタ値32、42等の値を変更して実行判断する時期を調整する必要がなく制御が簡単になる。上記のクロック数を保持したならば他の命令にジャンプする時点の減算値を退避させるなどの措置を要する。

【0144】

(4) 条件付きのディレイド分岐、あるいはジャンプ命令であれば、図11のようにクロックt6で分岐の条件を判断し、1クロックサイクルあるいはそれ以上のクロックサイクルを経過した後に開始するクロックt8で分岐を実行している。もし、サブ命令I41、I42がクロックt7でジャンプ命令DJMPの実行条件を決定するフラグF0を書き換える命令であるならば、サブ命令I01として実行条件の判断と同じクロックサイクルでジャンプを実行してしまうディレイなしのジャンプ命令が用いることができる。よって、本実施の形態のような条件付きのディレイド演算命令は、プログラムにおける命令スケジューリングをさらに柔軟にするものである。

【0145】

(5) 各演算ユニット、例えばPC制御部13は第1の演算命令の演算を実行

するようにPC制御部13を制御するための制御信号11、第1の演算命令の実行条件を示す第1の情報である条件指定フィールドの値、さらに、この実行条件を判断する時期を示す第2の情報であるアドレス値（あるいはそのアドレス値を保持するレジスタを指定する情報）をそれぞれレジスタ30、31、32に保持する。そして、これらレジスタの値は、第1の演算命令の後に命令デコーダ2でデコードされる第2の条件付き演算命令において、条件判断の時期を遅延させるために書き換えるまでそのまま保持し続ける。これは、図8のプログラムで条件付き演算命令により形成されるループ処理に効果的である。最初にレジスタ30、31、32に各情報を一旦保持させておけば、ループが発生する毎にそのレジスタ30、31、32に保持された情報を用いるだけで、条件付き演算命令が実行される。従来技術ならば、ループが生じる毎に条件付き演算命令を命令RAMからフェッチしてデコーダでデコードしなければならないが、本実施の形態によるとサブ命令I01の演算命令を1回だけフェッチし、デコードすればよい。従ってマイクロプロセッサの処理効率を上げることができる。

【0146】

(6) 本実施の形態では、マイクロプロセッサが処理するすべての命令は図2のフォーマットであり、そこには演算を指定するフィールドに対応して条件指定フィールドが設けられ、さらにこの条件指定フィールドに対応して条件判断遅延量指定フィールドが設けられている。条件指定フィールドは無条件で実行されることを指定することもできる。つまり、命令フォーマット中に条件付きか否かを示す情報、条件付きであればどのような条件か、その条件を判断する時期を示す情報を挿入する場所が確保されているので、プログラマー（あるいはコンパイラ）は容易に命令を構成することができる。

【0147】

(7) 条件判断遅延量指定フィールドにおいて、条件を判断する時期として命令デコードユニット2により命令のデコード段階で判断することを指定することもできる。命令の条件を満たせば命令実行部はその命令を無条件に実行される命令であるとして演算を行う。これも命令のスケジューリングに融通性を与える。

【0148】

(8) さらに次の変形例が考えられる。

条件付き演算命令の処理で条件を決定する際に参照された値はフラグ F0 であったが、これに限らず複数のビットを保持するレジスタの値であってもよい。

この実施例のマイクロプロセッサは VLIW アーキテクチャを採用して複数のパイプラインを並列に処理するものであるが、この条件付き演算命令で条件判断の時期を遅延させる機能は、単一のパイプラインを処理するプロセッサに設けても有効である。

【0149】

(9) 条件判断の時期を遅延させる手段である 3 つのレジスタを 1 セットにして各演算ユニットに設けているが、さらに各演算ユニットにこれら 3 つのレジスタと同じ機能をもつ 3 つのレジスタのセットを 1 又は複数個設けてもよい。これは、図 8 のプログラムで形成されたループの中に、さらに 1 又は複数のループが存在している場合に効果的である

【0150】

例えば、

100番地 : I01(BRA F0F #H30 110)

108番地 : I11(BRA F1T #H10 118)

110番地 : I21

118番地 : I31

120番地 : I41

128番地 : I51

130番地 : I61

というプログラムを考える。命令 I 0 1 は、フラグ F 0 が偽ならば 1 0 8 番地に分岐する分岐命令で、実行条件が 1 3 0 番地の命令 I 6 1 の命令実行ステージのサイクルで判断されて分岐が実行されるものである。命令 I 2 1 は、フラグ F 1 が真ならば 1 1 8 番地に分岐する分岐命令で、実行条件が 1 2 0 番地の命令 I 4 1 の命令実行ステージで判断されて分岐が実行される物である。このプログラムは、1 1 0 番地から 1 3 0 番地で第 1 のループが形成され、かつ 1 1 8 番地から

120番地で第2のループが形成されている。

【0151】

命令 I 01 がデコードされた場合、PC制御部 13 のレジスタ 30～32 に命令 I 01 の分岐命令に対する制御信号 11、条件を指定する情報、条件判断する時期を指定する情報を保持させる。命令 I 01 に次いで命令 I 11 がデコードされた場合、PC制御部 13 にさらに備えられた 3 つのレジスタに、命令 I 11 の分岐命令に対する制御信号 11、条件を指定する情報、条件判断する時期を指定する情報を保持させる。これにより、第1、第2のループが発生する毎にレジスタ 30、31、32 に保持された情報、同じ機能の 3 つのレジスタに保持された情報をそれぞれ参照するだけで適切に 2 つの条件付き演算命令が実行される。命令 I 01、I 02 をそれぞれループが発生する毎にフェッチし、デコードする必要はなくプロセッサの処理性能を上げることができる。

【0152】

実施の形態 2.

図 12 は図 1 のものとは別のマイクロプロセッサの構成を示したものである。

このマイクロプロセッサは、メモリ制御部 14 にレジスタ 43、記憶回路 44 を設け、ALU 15 にレジスタ 53、記憶回路 54 を設け、シフタ 16 にレジスタ 63、記憶回路 64 を設け、乗算機 17 にレジスタ 73、記憶回路 74 を設け、ALU 19 にレジスタ 83、記憶回路 84 を設け、シフタ 20 にレジスタ 93、記憶回路 94 を設けている。

その他の構成は、図 1 のものと同じである。

【0153】

図 1 のマイクロプロセッサにおいては、ディレイ機能を有する演算命令、つまり条件判断するクロックサイクルよりさらに後のクロックサイクルで演算を実行する演算命令は、分岐・ジャンプ命令のみであったが、ロードストア命令、算術演算演算、シフト算術演算命令等、任意の演算命令に対してディレイ機能を付加することはできる。各演算ユニットにさらに加えられた 6 つのレジスタは、PC制御部 13 のレジスタ 33 と全く同じ機能を有し、各演算ユニットで実行されるべき演算を実行する時期を示す情報を保持する。また各演算ユニットにさらに加

えられた6つの記憶回路は、PC制御部13の記憶回路34と全く同じ機能で、各演算ユニットで実行されるべき演算の実行条件の判断結果の情報を保持する。

【0154】

このように構成されたマイクロプロセッサは、例えば、

DADD F0T #H20, H30 R5,R5,R6

という命令を用意することができる。

この命令は、「フラグF0が真」を条件にしてレジスタR5、R6のそれぞれ内容を加算して、その加算結果をレジスタR5に書き込む加算命令である。この加算命令は、即値として2つのオフセット値‘20’‘30’を有する。加算命令DADDのアドレスが2000番地であれば、 $2000 + 20 = 2020$ 番地の演算命令の命令実行ステージE/Mが処理されるクロックサイクルで加算命令DADDの条件が判断される。そして $2000 + 30 = 2030$ 番地の演算命令の実行ステージE/Mが処理されるクロックサイクルで加算演算を実行する。この加算命令DADDが図2のoperation_0であれば、ALU15で条件が判断されて加算が行われ、operation_1であれば、ALU19で条件が判断されて加算が行われる。

【0155】

この実施の形態で各演算ユニットにレジスタ、記憶回路を付加したが、無論複数の演算ユニットの選択されたものにのみに備えてもよい。その場合にはそのレジスタ、記憶回路が付加された演算ユニットが処理できる演算命令のみにディレイ機能が付加されることになる。

【0156】

【発明の効果】

以上説明したように、この発明に係るデータ処理装置によると、第1の演算命令は第1の期間でデコードされ、それに続く第2の期間で実行される一方、所定の条件のもので演算が実行される第2の演算命令は、第3の期間でデコードされ、第3の期間が終了して少なくとも第2の期間と同じ時間を経過した第4の期間で条件が判断されて実行される。条件付きの演算命令がデコードされても、第1の演算命令のように直ちに実行するのではなくその実行条件を判断する時期を遅延

させた上で実行するので、その遅延している間に第2の演算命令の実行条件を決定する演算命令を実行することができる。よってプログラムシーケンスにおいて、第2の演算命令をその実行条件を決定する別の演算命令より前に記述することが可能で、命令のスケジューリングの自由度が増す。

【0157】

また、第2の演算命令で指定された演算が実行される時期を、条件判断が行われる時期よりさらに遅延させることにより、その遅延の間に、他の演算命令の処理で第2の演算命令の条件を決定している値が書き換えられても適切に条件付き演算命令を実行することができる。命令のスケジューリングの自由度がさらに増す。

【0158】

また、別のデータ処理装置によると、条件付き演算命令を処理する際に使用される第1ないし第3のレジスタを命令実行部に備えている。第1のレジスタは命令デコーダから出力される制御信号を保持し、条件付き演算命令に指定される演算が実行される条件を示す第1の情報を保持し、第3のレジスタはその条件の判断を開始する時期を示す第2の情報を保持するものである。

命令実行部がこれら3つの情報に基づき演算命令がデコーダされて、所定の時間を経過したのち条件が判断されて実行されるので、命令のスケジューリングにその遅延している間に第2の演算命令の実行条件を決定する演算命令を実行することができる。よって命令のスケジューリングの自由度が増す。

【図面の簡単な説明】

【図1】 この発明の実施の形態1によるマイクロプロセッサの構成を示すブロック図である。

【図2】 マイクロプロセッサの命令フォーマットを示す説明図である。

【図3】 マイクロプロセッサの並列2命令実行時のパイプライン動作を示す説明図である。

【図 4】 演算フィールドの詳細な内容を示す説明図である。

【図 5】 マイクロプロセッサのレジスタ構成を示す説明図である。

【図 6】 PSWの詳細内容を示す説明図である。

【図 7】 ディレイド分岐命令の基本的なフォーマットを示す説明図である。

【図 8】 図 1 のマイクロプロセッサで処理されるプログラムの一例を示す説明図である。

【図 9】 図 8 のプログラムを処理したときのマイクロプロセッサのパイプライン動作を示す説明図である。

【図 10】 図 1 のマイクロプロセッサで処理される別のプログラムの一例を示す説明図である。

【図 11】 図 9 のプログラムを処理したときのマイクロプロセッサのパイプライン動作を示す説明図である。

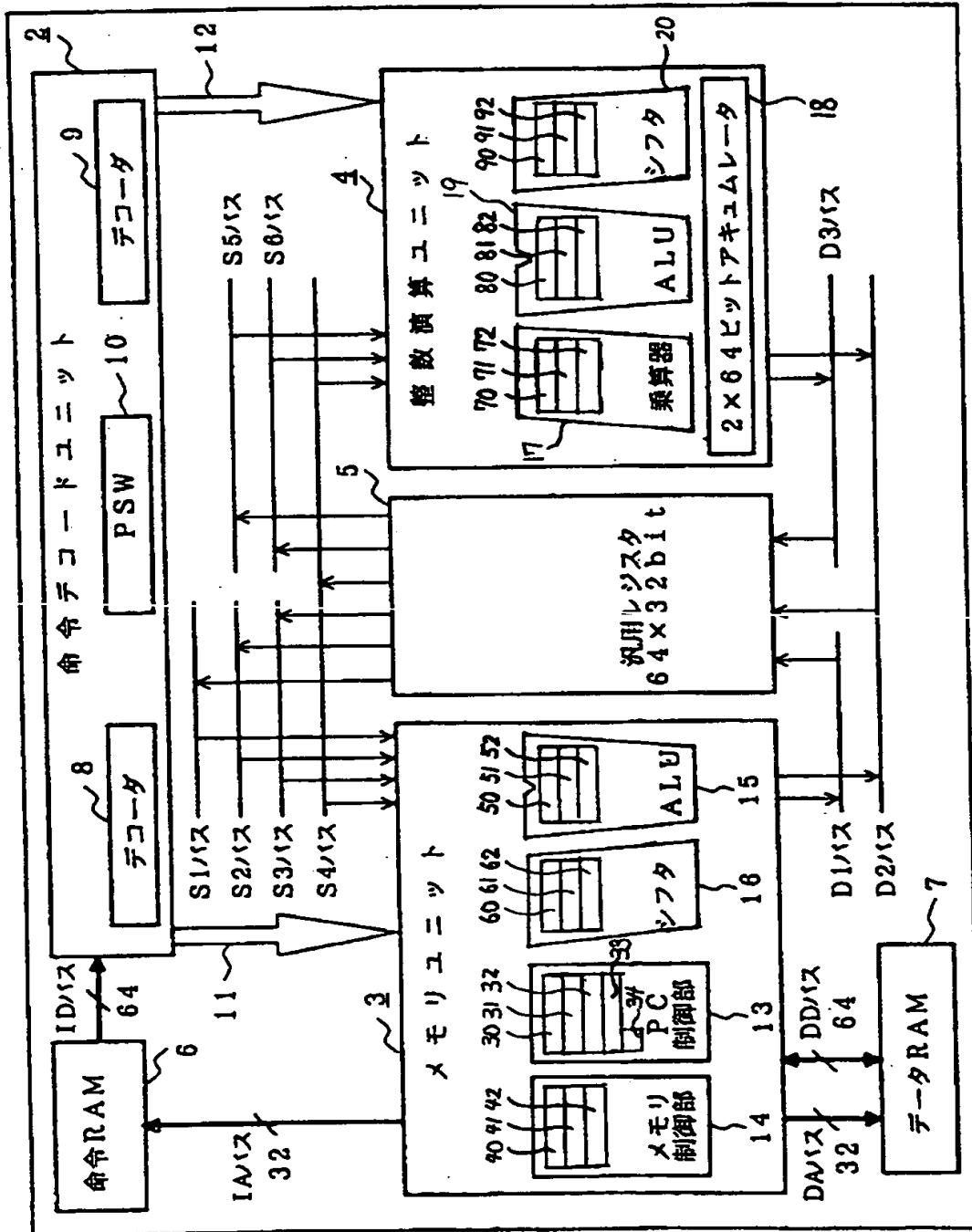
【図 12】 この発明の実施の形態 1 によるマイクロプロセッサの構成を示すブロック図である。

【符号の説明】

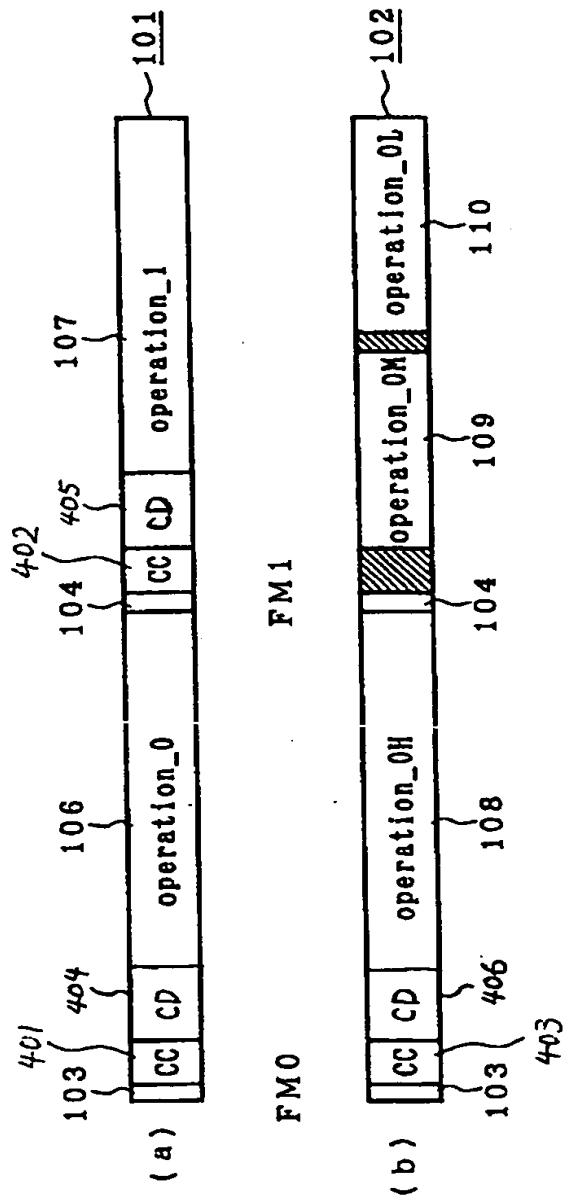
2…命令デコードユニット、3…メモリユニット、4…整数演算ユニット、5…汎用レジスタ、10…PSW、13…PC制御部、14…メモリ制御部、15、19…ALU、16、20…シフタ、17…乗算器、30～33、40～42、50～52、60～62、70～72、80～82、90～92…レジスタ、34…記憶回路

【書類名】 図面

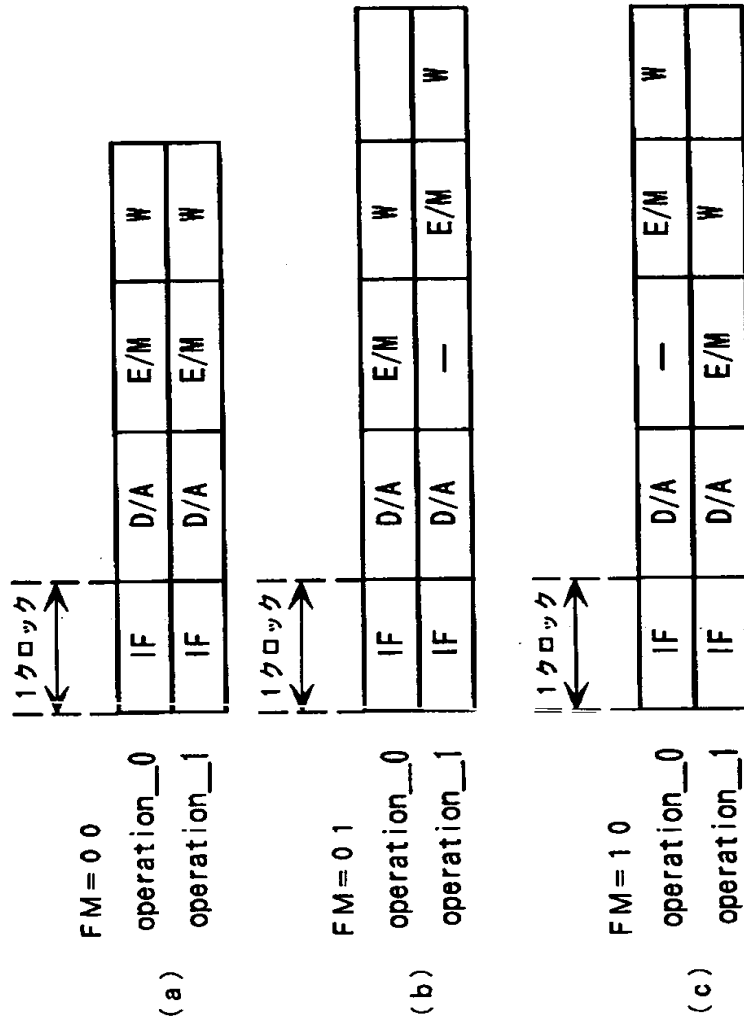
【図1】



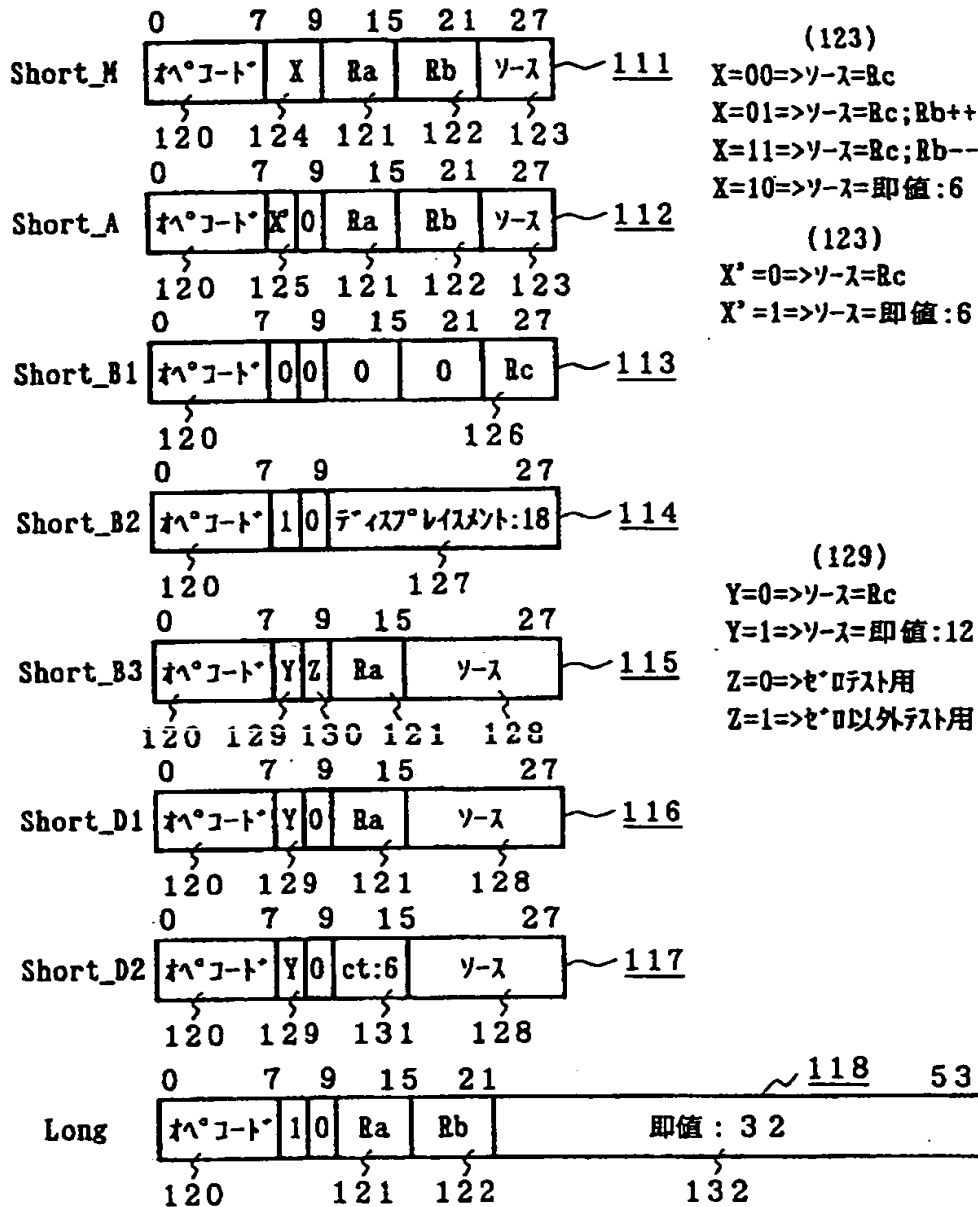
【図2】



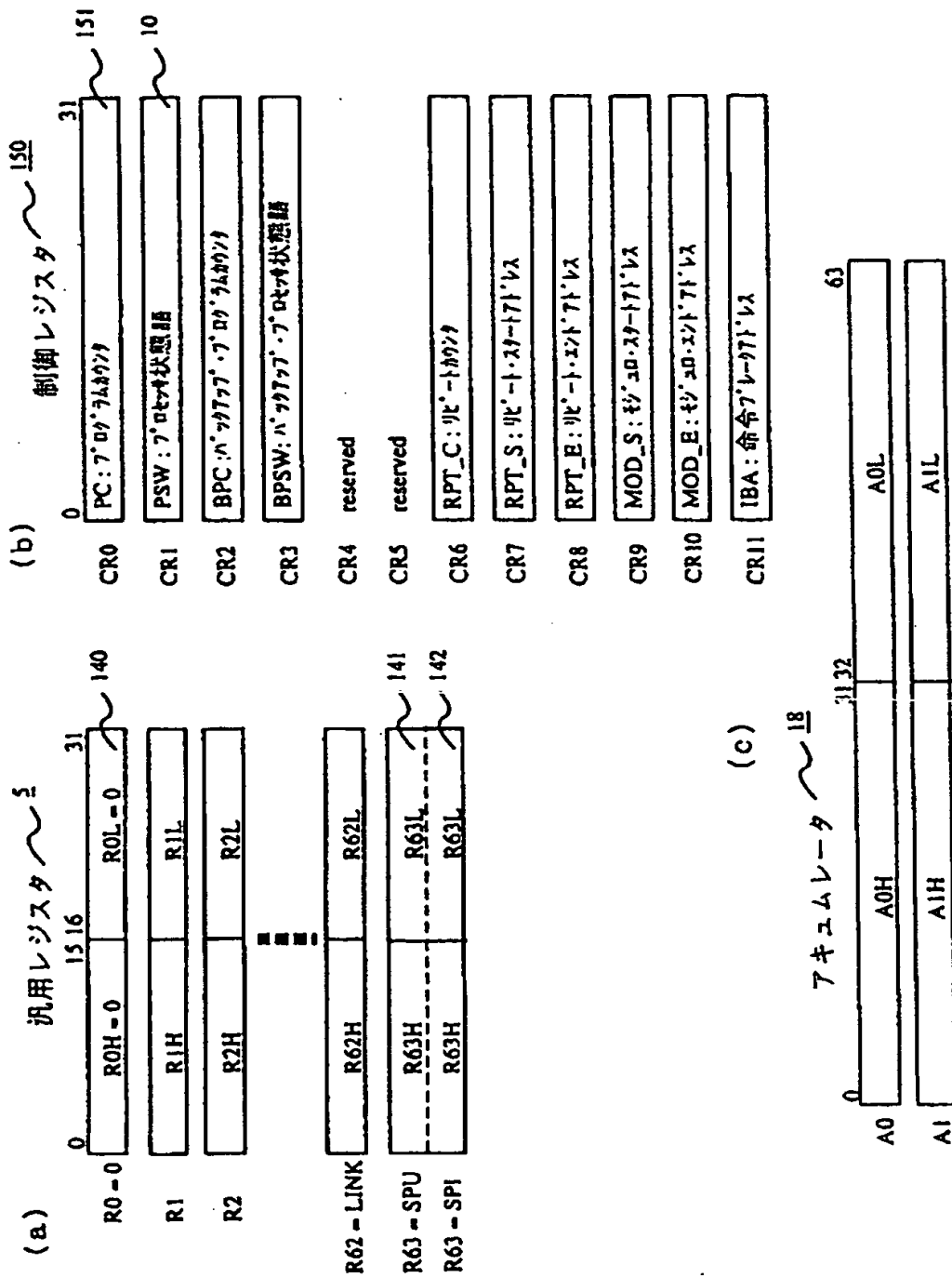
【図 3】



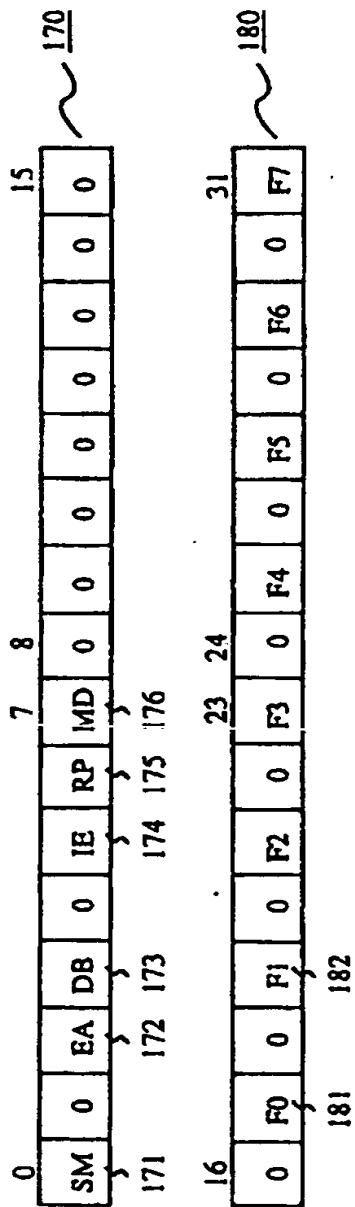
【図4】



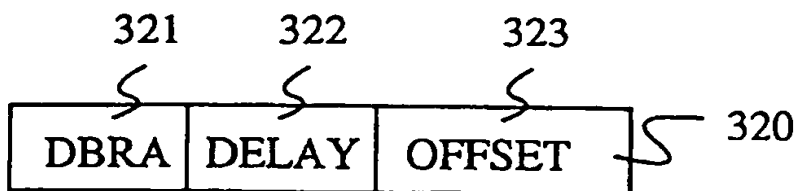
【図 5】



【図 6】



【図 7】



【図 8】

命令アドレス	operation_0	operation_1
H'1000	101 (BRA F0F #H'28 loop)	102 :
H'1008	loop: 111	112 :
H'1010	121 (ADD R2, R2, R3)	122 :
H'1018	131 (CMPEQ R2, R4, F0)	132 :
H'1020	end: 141	142 :
H'1028	151	152 :
H'1030	161	162 :

【図 9】

カウンタ値	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13
PCの値	H' 0FF0	H' 0FF8	H' 1000	H' 1008	H' 1010	H' 1018	H' 1020	H' 1008	H' 1008	H' 1008	H' 1010	H' 1018	H' 1020
101(BRA)	IF	D/A	E/M	E/M
102	IF	D/A	E/M	W	.	.	(条件判断)	(条件判断)
111		IF	D/A	E/M	W								
112		IF	D/A	E/M	W								
121(ADD)			IF	D/A	E/M	W							
122			IF	D/A	E/M	W							
131(CMPEQ)				IF	D/A	E/M	W						
132				IF	D/A	E/M	W						
141				IF	D/A	E/M	E/M	W					
142				IF	D/A	E/M	E/M	W					
151						IF	D/A	E/M	W				
152						IF	D/A	E/M	W				
161							IF	D/A	E/M	W			
162							IF	D/A	E/M	W			
111								IF	D/A	E/M	W		
112								IF	D/A	E/M	W		

【図 10】

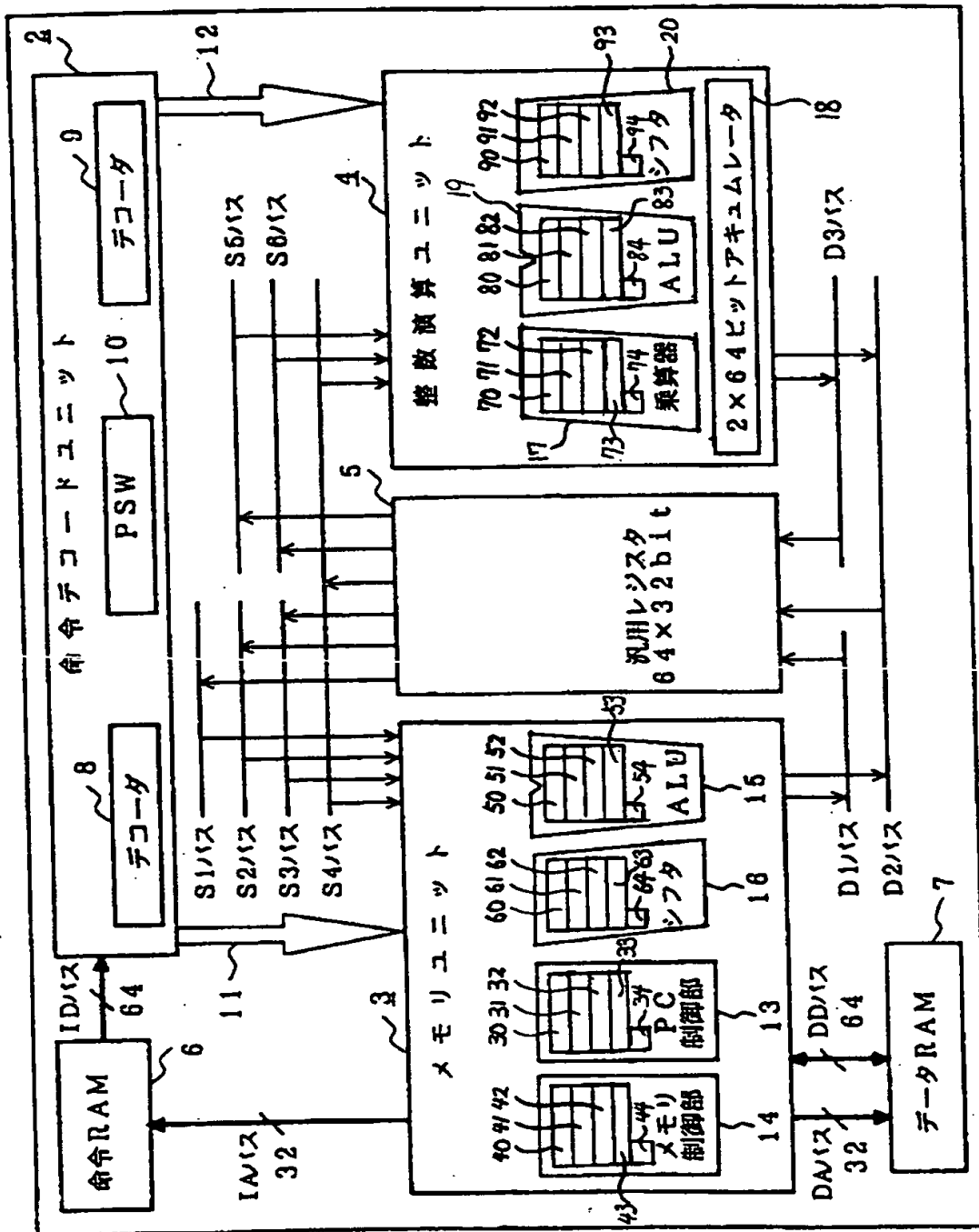
命令アドレス	operation_0	operation_1
H'1000	101 (DJMP F0T #H'18 #H'28 R5)	102 :
H'1008	111 (ADD R1, R1, R2)	112 :
H'1010	121 (CMPEQ R1, R3, F0)	122 :
H'1018	131 (ADD R5, R5, R6)	132 :
H'10	141	142 :
H'1028	151	152 :
H'1030	161	162 :
H'1038	171	172 :

【図 1 1】

カウンタ/命令	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
PCの値	H' 0FF0	H' 0FF8	H' 1000	H' 1008	H' 1010	H' 1018	H' 1020	H' 1028	H' 2000	H' 2000	H' 2000	H' 2008
101(DJMP)	IF	D/A	•	•	•	条件判断	•	E/M				
102	IF	D/A	E/M	W								
111(ADD)		IF	D/A	E/M	W							
112		IF	D/A	E/M	W							
121(COME0)												
122			IF	D/A	E/M	IF						
131(ADD)			IF	D/A	E/M	E/M	W					
132			IF	D/A	E/M	E/M	W					
141					IF	D/A	E/M	W				
142					IF	D/A	E/M	W				
151						IF	D/A	E/M	W			
152						IF	D/A	E/M	W			
161							IF	D/A	E/M	W		
162							IF	D/A	E/M	W		
171								IF	D/A	E/M	W	
172								IF	D/A	E/M	W	

2000番地の命令 {

【図12】



【書類名】 要約書

【要約】

【課題】 条件付き演算命令を含んだプログラムにおいて、命令のスケジューリングの自由度が向上させることのできるデータ処理装置を提供する。

【解決手段】 分岐の演算を実行するPC制御部13において、命令デコードユニット2でデコードされた結果を保持する第1のレジスタ30、演算の実行条件を示した情報（条件指定フィールドの値）を保持するレジスタ31、実行条件を判断する時期を示した情報（PCのアドレス値）を保持するレジスタ32を備えている。レジスタ32に保持された値がPC値と一致したときに、レジスタ31の情報に従って、実行条件の判断を開始する。もし条件を満たしていれば、レジスタ30に保持された内容に基づきPC制御部13が演算を実行する。実行条件を判断する時期を遅延させることができ、この遅延している間にその実行条件を決定するための演算を行える。従って分岐命令はプログラム上実行条件を決定する演算命令より前に置くことができる。

【選択図】 図1

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000006013
【住所又は居所】 東京都千代田区丸の内二丁目2番3号
【氏名又は名称】 三菱電機株式会社

【代理人】 申請人

【識別番号】 100102439
【住所又は居所】 東京都千代田区丸の内2丁目2番3号 三菱電機株式会社内

【氏名又は名称】 宮田 金雄

【選任した代理人】

【識別番号】 100103894
【住所又は居所】 東京都千代田区丸の内2丁目2番3号 三菱電機株式会社内

【氏名又は名称】 家入 健

【選任した代理人】

【識別番号】 100092462
【住所又は居所】 東京都千代田区丸の内2丁目2番3号 三菱電機株式会社内

【氏名又は名称】 高瀬 彌平

出 願 人 履 歴 情 報

識別番号 [000006013]

1. 変更年月日 1990年 8月24日
[変更理由] 新規登録
住 所 東京都千代田区丸の内2丁目2番3号
氏 名 三菱電機株式会社